سلسلة الدار المصرية للعلوم



تعلم بنفسك

Visual Basic.net

2005

محمد كامل عبد الحافظ



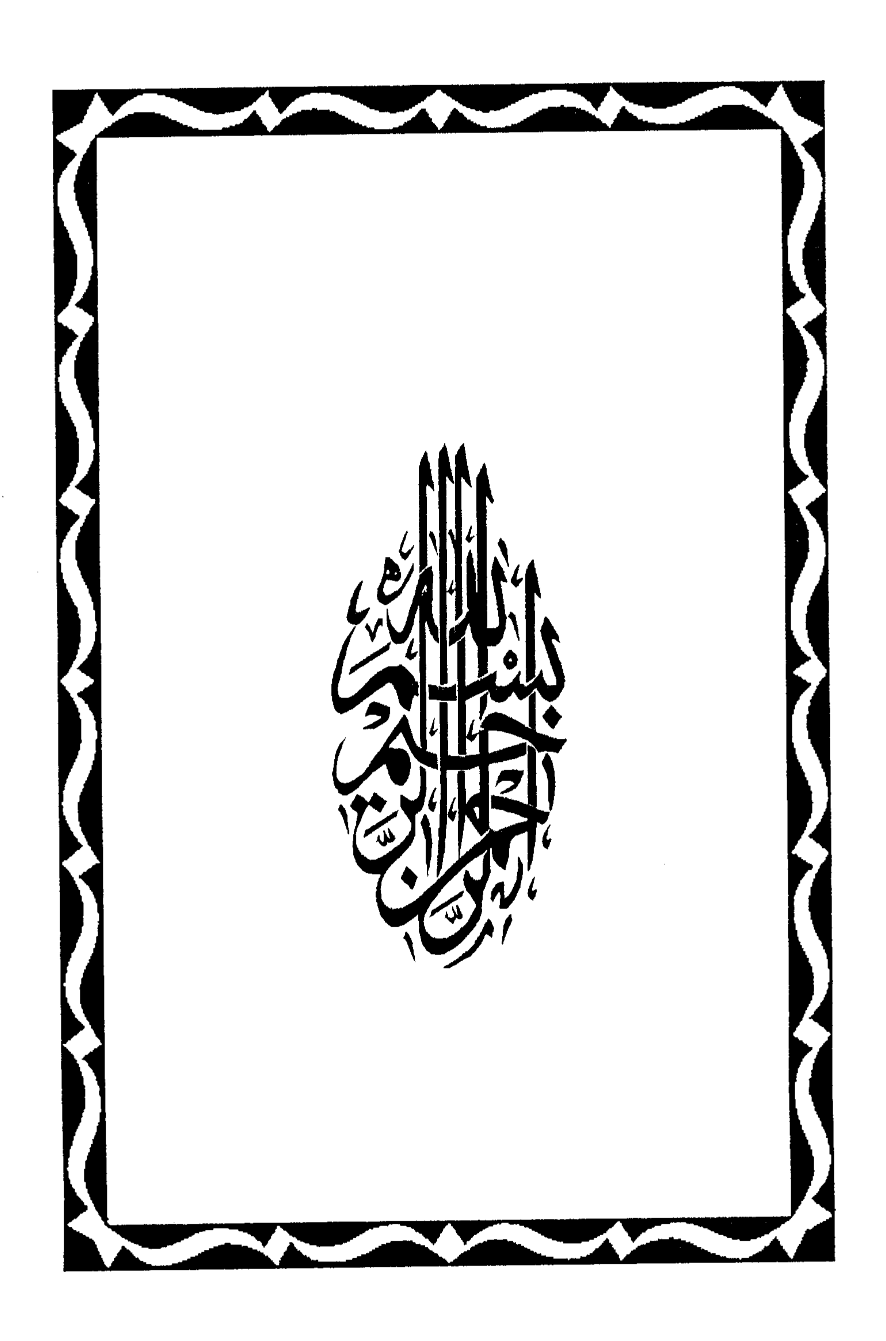
الكتـــــاب : **تعلم بنفسك** Visual Basic.net 2005
اعـــــداد : **محمد كامل عبد الحافظ**
المقــــاس : 21 X 15
الطبعــــة : الأولى
عدد الصفحات : 232
الناشـــــر : الدار المصرية للعلوم (نشر – توزيع)
رقم الايـداع : 2008 / 1932

الإخراج الفنى وتصميم الغلاف : جمال خليفة

بطاقة فهرسة

عبد الحافظ ، محمد كامل
Basic.net Visual2005
محمد كامل عبد الحافظ – القاهرة : الدار المصرية للعلوم ، 2008
ص ؟ سم .– (سلسلة تعلم بنفسك)
تدمك 5 36 6229 977 978
1– الحاسبات الالكترونية – برامج
2– Visual Basic.net 2005 (برامج حاسبات)
أ– العنوان
001.6425

الدار المصرية للعلوم
13 شـــارع اسـماعيـل أبو جبـل
خلف مستشفى الجمهورية – عابدين
☎ 23936478 – فاكس 23936079
elmasryabooks.net
seh_egypt@hotmail.com


بسم الله الرحمن الرحيم



مقدمة

يتناول هذا الكتاب شرح مبسط للغة فيجوال بيسيك دوت نت , و التي تعتبر من أهم منتجات شركة مايكروسوفت و التي يمكنها التعامل مع كل أنظمة تشغيل الويندوز , و تتميز لغة فيجوال بيسيك دوت نت باستخداماتها الواسعة و إمكانياتها المتعددة في التعامل مع العديد من أنواع قواعد البيانات , و إنشاء البرامج القابلة للتنفيذ سواء في جانب الويندوز أو للاستخدام في عمل تطبيقات الويب علي شبكة الإنترنت .

و يتناول هذا الكتاب شرح لغة فيجوال بيسيك دوت نت من حيث التعرف ببرنامج فيجوال أستوديو و إمكانيات هذا البرنامج في التعامل مع العديد من لغات البرمجة , و كيفية استخدام هذا البرنامج , و شرح لقوائم و نوافذ هذا البرنامج , بالإضافة إلي شرح مبسط لقواعد لغة فيجوال بيسيك , حيث يتناول الفصل الأول من هذا الكتاب ثلاثة أمثلة مبسطة للتعرف علي كيفية استخدام البرنامج , و يتناول الفصل الثاني شرح لواجهة برنامج فيجوال أستوديو , و يتناول الفصل الثالث شرح لقوائم البرنامج و عرض لكل الأوامر التي تحتوي عليها هذه القوائم و وظيفة كل أمر في هذه القوائم , و الفصل الرابع يعرض يشرح المكونات الأخرى لبرنامج فيجوال أستوديو مثل أنماط العرض المتاحة في البرنامج و كيفية التنقل بينها كما يشرح أيضاً أشرطة الأدوات الموجودة في



البرنامج , و يحتوي الفصل الخامس علي شرح لأدوات التحكم في النماذج الموجودة في برنامج فيجوال أستوديو من حيث كيفية أنشاء هذه الأدوات و كيفية التحكم في خواصها , و الأحداث التي تستخدم في تشغيل هذه الأدوات , و هيكل بناء المشروع , و الفصل السادس يشرح قواعد فيجوال بيسيك فيشرح هذا الفصل أنواع البيانات و المتغيرات و المعاملات و الجمل البرمجية و الفئات و الفصل السابع عبارة عن مشروع مبسط لشرح كيفية جمع البيانات من المستخدم و طرق جمع هذه البيانات المتاحة في فيجوال بيسيك دوت نت مثل مربعات الاختيار و قوائم الاختيار و مربعات الحوار , و الفصل الخامس يتناول موضوع التعامل مع قواعد البيانات من خلال برنامج فيجوال أستوديو .

و أرجوا أن أكون بهذا العرض المبسط قد قدمت مدخلاً لفيجوال بيسيك دوت نت بحيث يضعك هذا الكتاب علي أول الطريق لتعلم هذه اللغة , و لقد راعيت عند عرض الموضوعات التي يحتوي عليها هذا الكتاب أن يناسب المبرمج المبتدئ في التعامل مع لغات البرمجة بوجه عام , و لذلك لم أعتمد علي أي خلفية موجودة لدي القارئ , بحيث يعتبر هذا الكتاب مناسباً للقارئ المبتدئ في تعلم لغات البرمجة مع عدم الإخلال للمحتوي العلمي للغة .

شكر و تقدير

و أتوجه بالشكل لله عز وجل الذي وفقنا لأداء هذا العمل و أسأله أن ينفع به الناس و أن يجعله في ميزان أعمالنا يوم القيامة , و أتوجه بالشكر إلي أعزائنا القراء الذين حرصوا علي اقتناء هذا الكتاب للدخول إلي عالم المعرفة



بأحدي أهم لغات البرمجة , و أتوجه بالشكر لكل من ساهم في العمل في هذا الكتاب حتى وصل إلي أيدي أعزائنا القراء علي النحو الذي هو عليه , و أخص بالشكر العاملين في الدار المصرية للعلوم , و أخص بالشكر الأستاذ / محمد السيد باشا صاحب و مدير الدار , و الأستاذ /منصور مصباح محمد المدير التنفيذي للدار .

و أخيراً عزيزي القارئ يسعدني أن أتلقي استفسارات و توجيهاتك علي البريد الإليكتروني التالي mohamedkamel2004@yahoo.com حيث أستفيد كثيراً من رسائل القراء في التي تحمل تعليقاتهم و استفساراتهم و توجيهاتهم حيث أخذ هذه التوجيهات بعين الاعتبار في الإصدارات القادمة , أو يمكنك الاتصال بالمؤلف مباشرة علي الهاتف التالي .٠١١١٣٣٩٥٨٩

و الله و لي التوفيق .

المؤلف





أول مشروعات

باستخدام

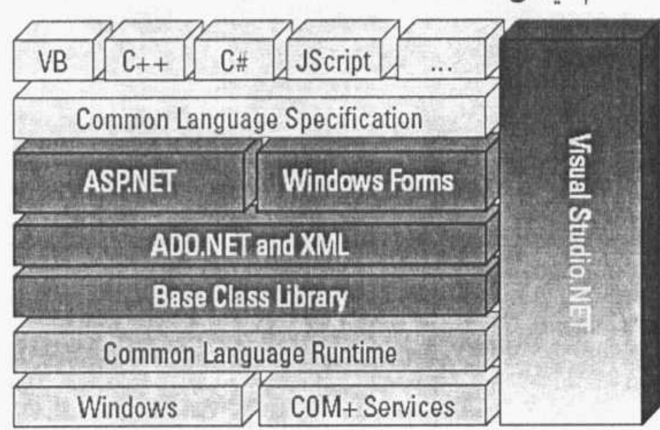
Visual Basic.net 2005

- المشروع الأول :
  - برنامج Hellow World
- المشروع الثاني :
  - برنامج Lucky7
- المشروع الثالث :
  - برنامج تاريخ الميلاد

المشروع الأول

برنامج Hellow World

قامت شركة مايكروسوفت بإنتاج بيئة دوت نت بحيث تتوافق مع كل إصدارات الويندوز ، و تضم بيئة دوت نت كما يوضحها (شكل رقم ١-١) العديد من لغات البرمجة أهمها لغة VB و لغة ++C و لغة #C و لغة Jscript والتي التي يمكن استخدام أي منها في برنامج واحد و هو فيجوال أستوديو و الذي يستخدم في أعداد الكثير من المشروعات البرمجية باستخدام أي من هذه اللغات



مكونات بيئة دوت نت
(شكل رقم ١-١)

و بتنصيب برنامج فيجوال أستوديو ، و اختيار أي من لغات البرمجة التي تريد استخدامها في عمل المشروع البرمجي الذي تقوم بعمله من بين لغات



البرمجة المتاحة في البرنامج .

برنامج فيجوال أستوديو

لكي تستخدم برنامج فيجوال أستوديو يجب الحصول علي البرنامج و تنصيبه علي الجهاز أولاً ، و هناك عدة إصدارات من برنامج فيجوال أستوديو ٢٠٠٥ كل إصدار من هذه الإصدارات مصمم بحيث يناسب نوع معين من المستخدمين أو نوع معين من العمل الذي يقوم به المستخدم .

فهناك إصدار للبرنامج يسمي Team System هذا الإصدار مخصص لإنشاء المشروعات البرمجية التي يتم إنشاؤها بواسطة فريق عمل كما يحدث في إنشاء المشروعات البرمجية الكبيرة في شركات البرمجة الكبيرة ، و يحتوي هذا الإصدار من برنامج فيجوال أستوديو علي عدد كبير من الأدوات التي تصلح للعمل في البرنامج كفريق ، و هناك إصدار أخر من برنامج فيجوال أستوديو يسمي Professional Eddition هذا الإصدار خاص للتعامل مع البرنامج من خلال شخص واحد فقط أو عدد قليل من الأشخاص ، و هذا الإصدار هو الإصدار الشائع للبرنامج و يستخدم في إنشاء المشروعات البرمجية المتوسطة و الصغيرة ، و هناك إصدار ثالث من برنامج فيجوال أستوديو يسمي Standard Editin هذا الإصدار مخصص لإنشاء المشروعات البرمجية الصغيرة و يحتوي علي ٨٠٪ فقط من إمكانيات الإصدار Professonal Edition و هناك إصدار رابع من برنامج فيجوال أستوديو يسمي Express Edition هذا الإصدار مخصص للمتدربين علي استخدام البرنامج و الهواة و لا يحتوي هذا الإصدار علي

العديد من أنواع المشروعات البرمجية التي تحتوي عليها الإصدارات الأخرى من البرنامج .

و يمكنك الحصول علي الإصدار الذي تريده من إصدارات فيجوال أستوديو من موقع شركة مايكروسوفت علي شبكة الإنترنت http://www.microsoft.com و ينصح أيضاً بالحصول علي مكتبات Microsoft's Handy Technical Archive و التي يمكنك الحصول عليها أيضاً من نفس الموقع

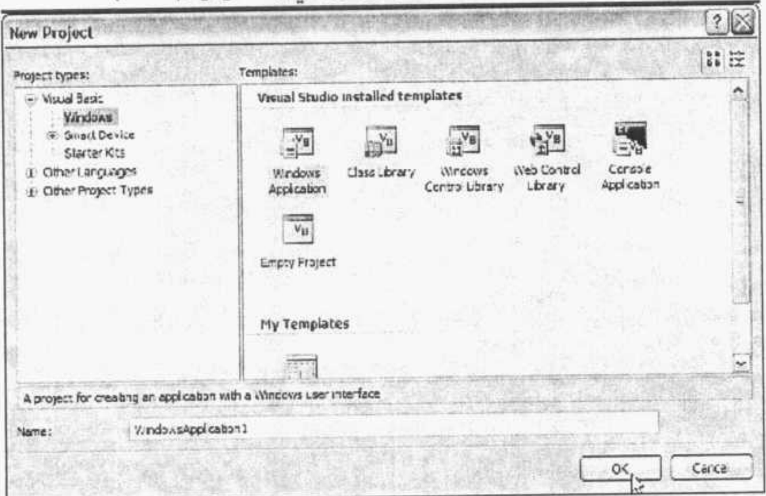
و عند قيامك بتنصيب برنامج فيجوال أستوديو لأول مرة ينصح باختيار الأعدادات الافتراضية للبرنامج للتعامل معه كمبتدئ , و بعد تنصيب البرنامج يمكنك الوصل أليه بالضغط علي قائمة Start الموجودة بأسفل يسار سطح المكتب , و اختيار All Programs و منها اختار Microsoft Visual Studio 2005 ليتم بذلك فتح البرنامج , ويمكنك بداية العمل مع البرنامج في إنشاء مشروع جديد باختيار New من قائمة File و منها اختار Project ثم قم باختيار اللغة التي تريد أنشاء المشروعات الجديد بها من بين لغات البرمجة المتوفرة في البرنامج .

و فيما يل سوف نوضح كيفية البدء في عمل مشروع جديد باستخدام فيجوال بيسيك و المشروع عبارة عن نموذج .

المشروع الأول باستخدام فيجوال بيسيك

في برنامج فيجوال أستوديو باختيار New من قائمة File و منها اختيار Project يظهر بذلك المربع الحواري الخاص باختيار لغة البرمجة التي تريد

إنشاء المشروع الجديد بها كما هو موضح في (شكل رقم ١-٢)



المربع الحواري الخاص باختيار لغة البرمجة المستخدمة في إنشاء المشروع
(شكل رقم ١-٢)

و في قائمة أنواع المشروعات Project Types الموجودة بيسار المربع الحواري الخاص بإنشاء مشروع جديد الموضح في الشكل السابق تظهر قائمة بلغات البرمجة التي يمكن استخدامها مع فيجوال أستوديو و في النافذة الموجودة في اليمين تظهر أنواع المشروعات التي يمكن أنشاؤها باستخدام أي لغة تقوم باختيارها من قائمة أنواع المشروعات .

من قائمة أنواع المشروعات Project Type الموجودة في يسار المربع الحواري اختار Visual Basic و قم بالنقر فوق علامة + الموجودة بجوار أسم اللغة و من القائمة التي تظهر أختار Window و من نافذة أنواع المشروعات

الموجودة في يمين المربع الحواري اختار Window Application و في حقل Name الموجود بأسفل المربع الحواري قم بكتابة أسم المشروع الذي تريد إنشاؤه , و ليكن علي سبيل المثال Hello World و أضغط زر Ok بأسفل المربع الحواري ليقوم فيجوال بيسيك بإنشاء نموذج يسمي Form1 و يعرض هذا النموذج في نافذة التصميم , و يقوم فيجوال أستوديو بعرض النموذج كاملاً بما في ذلك شريط العنوان الخاص بالنموذج و مقابض تغيير حجم النموذج التي تظهر حول النموذج و الزر الخاص بإغلاق النموذج و كافة الأدوات التي تمكنك من التعامل مع النموذج بالشكل الذي تريده في نافذة التصميم .

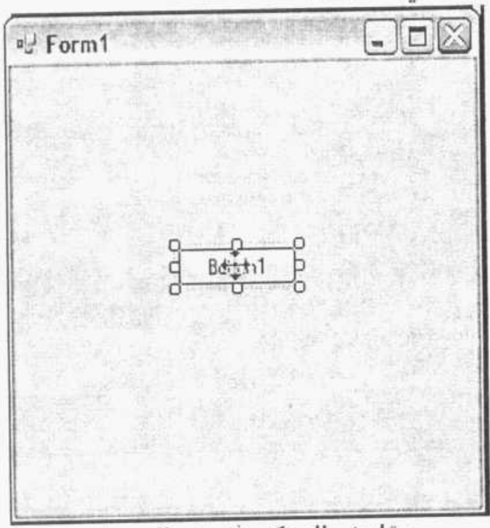
و في الجانب الأيسر من الشاشة قم بالنقر فوق كلمة Toolbox ليظهر بذلك شريط الأدوات الذي يحتوي علي أدوات التحكم في النموذج كما يوضحها (شكل رقم ١-٣) .

Toolbox
All Windows Forms
Common Controls
Pointer
Button
CheckBox
CheckedListBox
ComboBox
DateTimePicker
Label
LinkLabel
ListBox
ListView
MaskedTextBox
MonthCalendar
NotifyIcon
NumericUpDown
PictureBox
ProgressBar
RadioButton
RichTextBox
TextBox
ToolTip
TreeView
WebBrowser
Containers
Pointer
FlowLayoutPanel
GroupBox
Panel
SplitContainer

شريط أدوات التحكم في النموذج
(شكل رقم ١-٣)

في شريط أدوات التحكم في النموذج الموضح في الشكل السابق أنقر

نقراً مزدوجاً علي زر Button الموجود بأعلى القائمة كما يوضحه الشكل ليقوم فيجوال أستوديو بإضافة الزر إلي النموذج , و في النموذج قم بالنقر فوق مقابض التحكم في الزر كما يوضحها (شكل رقم ١-٤) , و قم بسحب الزر إلي المكان الذي تريده في النموذج .



مقابض التحكم في زر النموذج
(شكل رقم ١-٤)

لقد قمت الآن بإنشاء نموذج بسيط يحتوي هذا النموذج علي زر واحد , و يمكنك تحديد ما يحدث عندما يقوم المستخدم بالنقر علي هذا الزر , و يمكنك أيضاً عرض الأكواد التي أستخدمها البرنامج في إنشاء النموذج و إنشاء الزر كما يمكنك استخدام الأكواد في تحديد ماذا يحدث عندما يقوم المستخدم بالنقر علي هذا الزر .

تغيير أسم الزر

و لتغيير أسم الزر قم بالنقر فوق الزر في النموذج نقراً مزدوجاً ليقوم فيجوال أستوديو بعرض الأكواد المتعلقة بهذا الزر كما يلي .

```
• Public Class Form1
• Private Sub Button1_Click(ByVal sender As System.Object,
• ByVal e As System.EventArgs) Handles Button1.Click
• End Sub
• End Class
```

و لتغيير أسم الزر قم بتغيير السطر السابق لسطر End Sub إلي ما يلي :

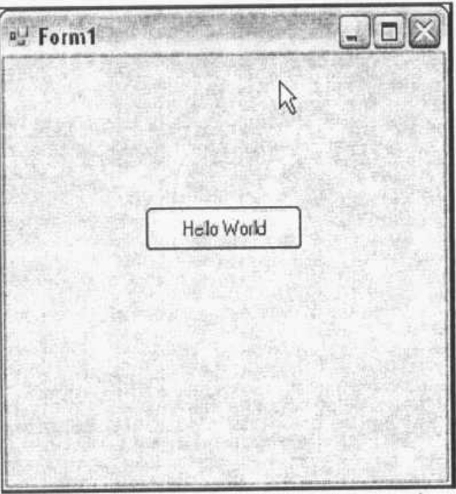
```
• Button1.Text = "Hello World"
```

بحيث يصبح الكود علي النحو التالي :

```
• Public Class Form1
• Private Sub Button1_Click(ByVal sender As System.Object,
• ByVal e As System.EventArgs) Handles Button1.Click
• Button1.Text = "Hello World"
• End Sub
• End Class
```

بعد ذلك أضغط مفتاح F5 في لوحة المفاتيح ليقوم فيجوال أستوديو بفتح المشروع الحالي كنافذة مشروع للتجربة كما يوضحها (شكل رقم ١-٥) .

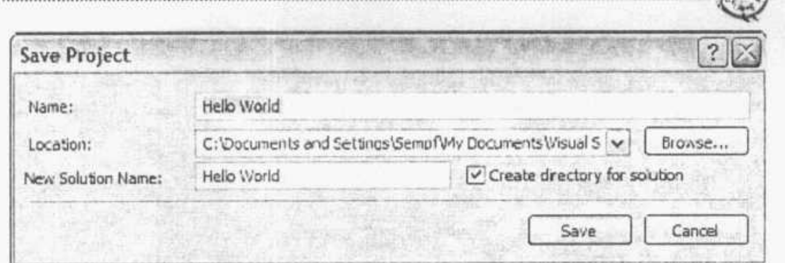




فتح المشروع في الشكل النهائي له

(شكل رقم ١-٥)

و لحفظ المشروع قم بإغلاقه أولاً بالنقر علي زر الإغلاق × الموجود في أعلي يسار النموذج , و اختار Save All من قائمة File الموجودة في شريط القوائم الخاص ببرنامج فيجوال أستوديو ليظهر بذلك المربع الحواري الخاص بحفظ المشروع كما يوضحه (شكل رقم ١-٦) , و الذي من خلاله تستطيع تحديد أسم المشروع و المكان الذي تقوم بحفظ المشروع فيه , و في هذا المربع الحواري قم بتحديد أسم المشروع و المكان الذي تريد حفظه فيه وأضغط زر Save الموجود في أسفل المربع الحواري ليتم حفظ المشروع و إغلاق المربع الحواري .



المربع الحواري الخاص بحفظ المشروع

(شكل رقم ١-٦)

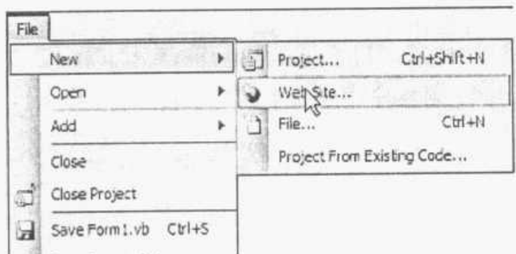
و لاستكمال بناء المشروع أختار Buid HelloWorld من قائمة Build ليقوم فيجوال أستوديو بتفسير الأكواد الخاصة بالمشروع و إخراجه في شكل قابل للتنفيذ في شكل ملف بامتداد exe. و يمكن الوصول إلي هذا الملف و فتحه و نسخه إلي المكان الذي تريده علي الجهاز من الملف الافتراضي الذي يقوم فيجوال أستوديو بحفظ المشروعات النهائية فيه و الذي مساره كالتالي :

- C:\Documents and Settings\sempf\
- My Documents\Visual Studio\Projects\Hello World\Hello World\
- bin.

- ثم أنقر نقراً مزدوجاً فوق البرنامج الذي قمت بإنشائه لتشغيله .

و لا يقتصر عمل فيجوال بيسيك علي أنشاء النماذج فقط , و لكن تستطيع فيجوال بيسيك أيضاً عمل برامج كاملة يمكن تشغيلها في بيئة ويندوز , كما يمكن لفيجوال بيسيك أيضاً عمل تطبيقات الويب التي تستخدم في صفحات الويب الديناميكية و التعامل مع قواعد البيانات , و كما يوضح (شكل رقم١-٧) أنه باختيار New من قائمة File في برنامج فيجوال أستوديو تعرض القائمة الفرعية New ضمن الخيارات الموجودة في القائمة خيار Web Site





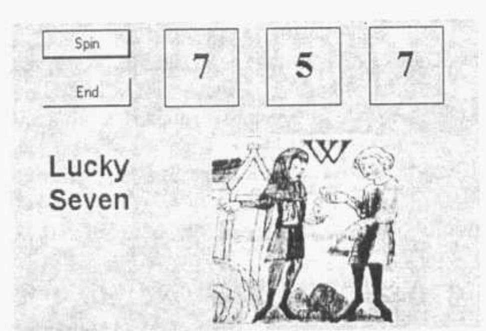
الخيار الخاص بإنشاء صفحات الويب باستخدام فيجوال بيسيك

(شكل رقم ١-٧)

المشروع الثاني

برنامج Lucky 7

في هذا المشروع سوف تقوم بإنشاء لعبة تسمي Lucky Seven و هذه اللعبة لها واجهة مستخدم كالموضحة في (شكل رقم ١-٨) , و تحتوي واجهة المستخدم كما يوضحها الشكل علي زرين , و ثلاثة أرقام هي أرقام الحظ , و صورة .



واجهة المستخدم الخاصة بالبرنامج

(شكل رقم ١-٨)

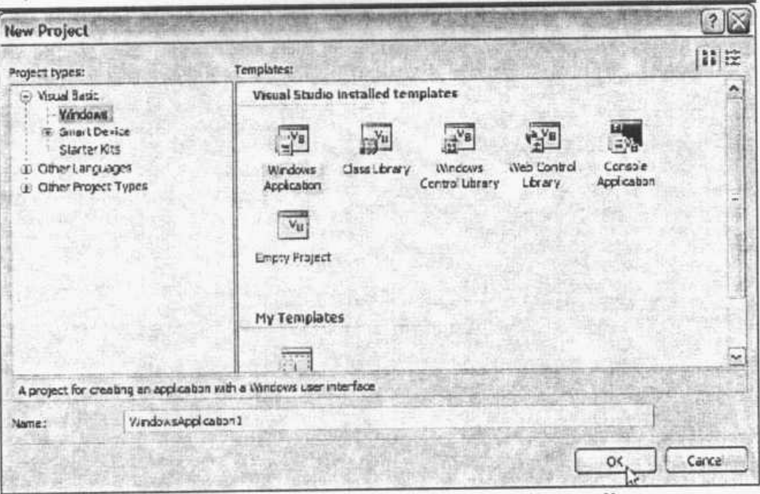
و سوف يتم إنشاء هذه اللعبة باستخدام سبعة نماذج و تم تغيير خواص كل نموذج من هذه النماذج , و إنشاء زر بقيام المستخدم بالنقر عليه يتم اختيار أحد هذه النماذج بطريقة عشوائية , و تتمثل الخطوات الخاصة بإنشاء هذه اللعبة في ثلاثة خطوات هي كما يلي :

- إنشاء واجهة المستخدم .
- تحديد خواص واجهة المستخدم .
- كتابة الكود الخاص بالبرنامج .

الخطوة الأولي : إنشاء واجهة المستخدم

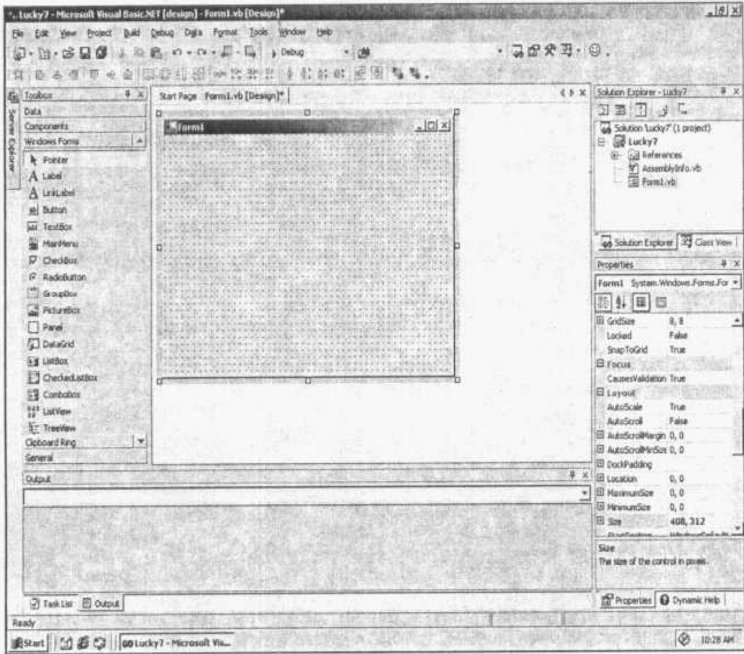
في هذه الخطوة قم بفتح برنامج فيجوال أستوديو , و في البرنامج أختار New من قائمة File و منها اختار Project ليظهر بذلك المربع الحواري الخاص بإنشاء مشروع جديد , و في هذا المربع الحواري قم بالنقر علي أيقونة Window Application في نافذة أنواع المشروعات الموجودة في يمين المربع الحواري كما يوضح (شكل رقم ١-٩) و في المربع الحواري الخاص بإنشاء نافذة تطبيقات قم بكتابة Mylucky7 في حقل Name ليكون هذا هو اسم البرنامج الذي تقوم بإنشائه , و اختار مسار حفظ البرنامج في المسار الافتراضي و أضغط زر Ok في أسفل المربع الحواري ليقوم فيجوال أستوديو بإنشاء ملف جديد خاص بالبرنامج الذي تقوم بإنشائه .





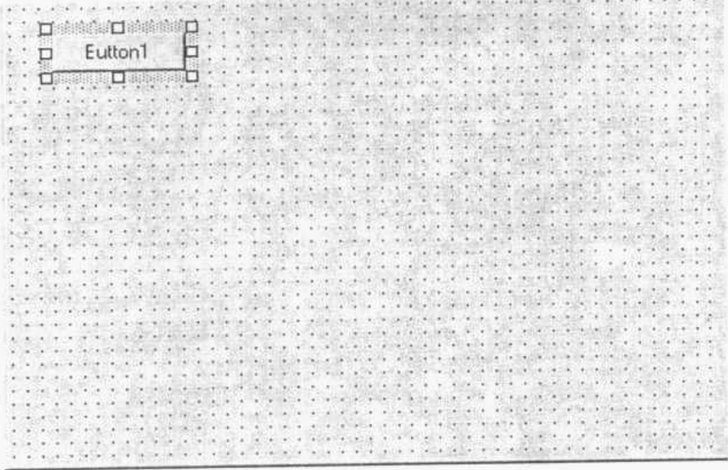
المربع الحواري الخاص بإنشاء مشروع جديد
(شكل رقم ١-٩)

بالضغط علي زر Ok في المربع الحواري الخاص بإنشاء نموذج الموضح في الشكل السابق يقوم فيجوال أستوديو بعرض نافذة تصميم النماذج Windows Form Designer و يظهر فيها النموذج بحيث تستطيع التعامل معه بشكل مرئي كما يوضح (شكل رقم ١-١٠) و في محرر النماذج الموضح في الشكل قم بتحريك مؤشر الماوس إلي الركن الأيمن السفلي للنموذج حتى يتحول شكل مؤشر الماوس إلي شكل سهمين متقابلين ثم قم بسحب طرف النموذج الأيمن السفلي برز الماوس إلي الخارج لتكبير حجم النموذج حتى يصبح تقريباً بحجم النموذج الموضح في الشكل .



نافذة محرر النماذج
(شكل رقم ١-١٠)

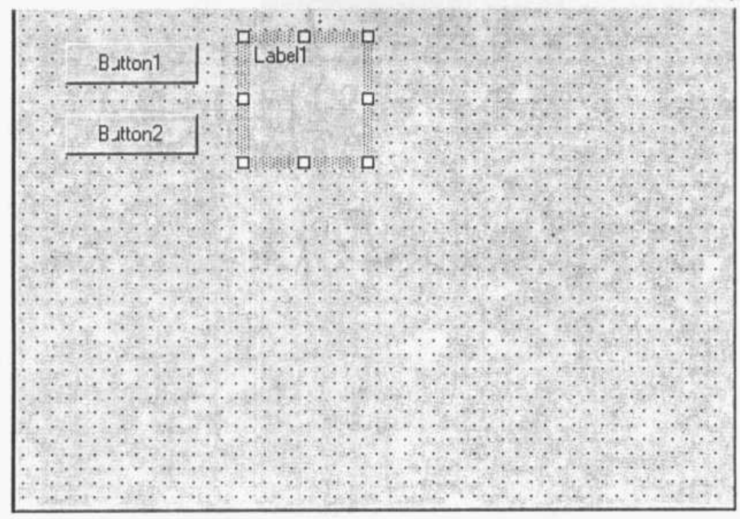
بعد إنشاء النموذج كما يوضح الشكل السابق , قم بالنقر علي زر Button الخاص بإضافة الأزرار للنموذج الموجود في شريط الأدوات Toolbox في يسار واجهة البرنامج , ثم قم بتحريك مؤشر الماوس فوق النموذج ليتحول بذلك شلك مؤشر الماوس و يأخذ شكل علامة + قم بالسحب بزر الماوس فوق النموذج لرسم الزر فوق النموذج في المكان الذي تريده , كما يوضح (شكل رقم ١-١١)



إدراج زر النموذج
(شكل رقم ١-١١)

و يمكنك تعديل حجم الزر و مكانه بسحبه من مقابض التحكم البيضاء التي تظهر حول الزر حتى تحصل علي شكل الزر و حجمه و مكانه الذي تريده , و بنفس الطريقة قم بإضافة زر أخر إلي النموذج أسفل الزر الأول .

و بنفس الطريقة التي قمت بإضافة الأزرار إلي النموذج قم بإضافة مربع عنوان Label إلي النموذج كالموضح في (شكل رقم ١-١٢) , و ذلك بالنقر فوق Label في شريط الأدوات الموجود في يسار واجهة البرنامج ليتحول بذلك شكل مؤشر الماوس إلي شكل علامة + قم بالسحب بزر الماوس فوق النموذج لإضافة مربع العنوان كما يوضح الشكل .



إضافة مربع عنوان
(شكل رقم ١-١٢)

و يمكنك بعض إضافة مربع العنوان تغيير شكل و مكان و حجم مربع العنوان عن طريق مقابض التحكم الموجودة خلف مربع العنوان , و بنفس الطريقة قم بإضافة مربعات عنوان أخري إلي النموذج كما يوضحها (شكل رقم ١-١٣) .



إضافة مربعات عنوان أخري إلي النموذج

(شكل رقم١-١٣)

إضافة صورة إلي النموذج

لإضافة صورة إلي النموذج قم أولاً بإضافة المربع الخاص بوضع الصورة كما هو موضح في (شكل رقم ١-١٤) , و ذلك بالنقر فوق PictureBox في شريط الأدوات الموجود في يسار واجهة البرنامج ثم قم بالسحب فوق النموذج لإضافة المربع إلي النموذج كما يوضح الشكل .

المربع الخاص بإضافة الصورة

(شكل رقم ١-١٤)

الخطوة الثانية : تحديد خواص واجهة المستخدم

لقد قمت بإضافة ثلاثة أنواع من أدوات التحكم في النموذج هي الأزرار و مربعات العناوين و مربع الصورة , و فيما يلي سوف نقوم بتحديد خواص كل هذه الأدوات .

تحديد خواص الأزرار

قم بالنقر الزر الأول الموجود بأعلى يسار النموذج لتحديده , و ظهور مقابض التحكم حول هذا الزر , ثم أنقر نقراً مزدوجاً فوق شريط العنوان الخاص بنافذة الخواص Properties Window و أن لم تكن هذه النافذة

ظاهرة في واجهة البرنامج قم بإظهارها بالضغط علي مفتاح F4 في لوحة المفاتيح كما يوضحها (شكل رقم ١-١٥) .

Properties

btnMakeHiButton System.Windows.Forms.Button

Property	Value
ImageAlign	MiddleCenter
ImageIndex	(none)
ImageKey	(none)
ImageList	(none)
⊞ Location	0, 0
Locked	False
⊞ Margin	3, 3, 3, 3
⊞ MaximumSize	0, 0
⊞ MinimumSize	0, 0
Modifiers	Friend
⊞ Padding	0, 0, 0, 0
RightToLeft	No
⊞ Size	75, 23
TabIndex	0
TabStop	True
Tag	
Text	Make Hi Button
TextAlign	MiddleCenter
TextImageRelation	Overlay
UseCompatibleTextRende	False
UseMnemonic	True
UseVisualStyleBackColor	True
UseWaitCursor	False
Visible	True

Text
The text associated with the control.

نافذة الخواص

(شكل رقم ١-١٥)

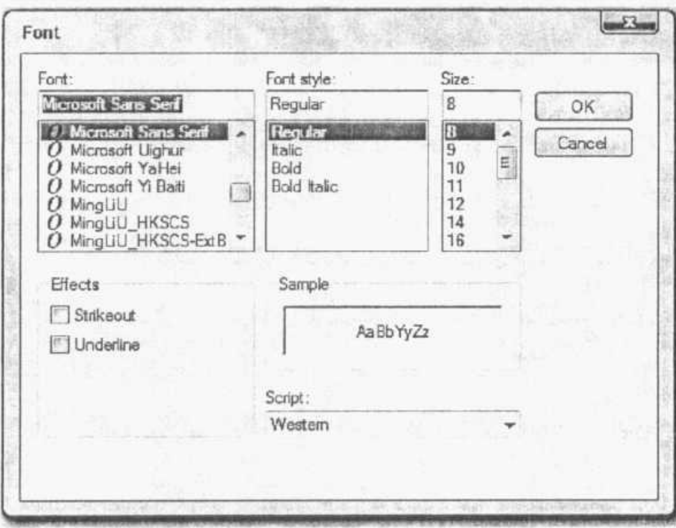
و يظهر بقائمة الخواص خواص العنصر المحدد من بين عناصر التحكم في النموذج و هو الزر Button1 و تتضمن هذه القائمة الخواص المتعلقة بهذا الزر مثل لون الخلفية Background Color و النص المكتوب علي الزر Text و الخط المستخدم في كتابة هذا النص Font و حجم الزر العرض و الارتفاع Width و Height و العديد من الخواص الاخري للزر , و تقوم هذه النافذة بتجميع الخواص المتعلقة بالزر في مجموعات , و يظهر بجوار كل مجموعة من مجموعات الخواص الموجودة في القائمة علامة + يسار أسم المجموعة و بالنقر علي هذه العلامة تظهر كل الخواص الموجودة في هذه المجموعة .

و لتغيير النص المكتوب فوق الزر أنقر نقراً مزدوجاً فوق الخاصية Text في القائمة ليتم تحديد النص في النافذة بالطريقة التي تسمح لك بتعديل هذا النص , قم باستبدال النص بعبارة Spin و أضغط مفتاح Enter في لوحة المفاتيح ليتم بذلك تعديل النص المكتوب علي الزر , و بنفس الطريقة قم بتغيير النصر المكتوب فوق الزر الثاني Button2 ليصبح End

تغيير خواص مربعات العناوين

بنفس الطريقة التي قمت بتغيير خواص الأزرار بها يمكنك تغيير الخواص الخاصة بمربعات العناوين , قم بالنقر فوق مربع العنوان Lable1 في النموذج لتحديده , ثم قم بالضغط علي مفتاح Shift في لوحة المفاتيح و النقر علي مربع العنوان الأخر Lable2 و مربع العنوان الثالث Lable3 لتحديد الثلاثة مربعات معاً , ثم قم بالنقر فوق خاصية TextAlign الخاصة بتحديد اتجاه

النص في نافذة الخواص Properties Window و في هذه الخاصية قم بالنقر علي السهم الأسود الصغير الموجود بيسار الخاصية لظهور الاختيارات الخاصة بهذه الخاصية و المتعلقة بتحديد اتجاه النص المكتوب في مربعات العناوين, من هذه الخيارات أختار MiddleCenter لتوسيط النص الموجود في مربعات العناوين الثلاثة المحددة في النموذج , و الثلاثة مربعات مازالت محددة اختار خاصية BorderStyle من نافذة الخواص و المتعلقة بشكل الإطار المحيط بالمربع , و في هذه الخاصية قم بالنقر أيضاً علي السهم الأسود الصغير الموجود بيمين الخاصية لظهور الخيارات المتعلقة بهذه الخاصية من هذه الخيارات أختار FixedSingle و المربعات الثلاثة مازالت محددة اختار خاصية Font من نافذة الخواص و قم بالنقر فوق الزر الدائري الموجود بجوار هذه الخاصية ليظهر بذلك المربع الحواري الخاص بتحديد حجم الخط و نوعه و تنسيقه كما هو موضح في (شكل رقم ١-١٦) , و من هذا المربع الحواري أختار نوع الخطTimes New Roman و تنسيق الخط Bold و حجم الخط 24 و أضغط زر Ok الموجود بأعلى المربع الحواري .



المربع الحواري الخاص بتنسيق الخط

(شكل رقم ١-١٦)

و الآن ستقوم بإلغاء النص الافتراضي المكتوب في مربعات العناوين و هو Label1 و Lable2 و Label3 و Lable4 و بإلغاء النصوص من هذه المربعات لن يتم إلغاء تنسيق النصوص الخاص بهذه المربعات الذي قمت بإعداده بل أن أي نص تقوم بكتابته في هذه المربعات بعد ذلك سوف يطبق عليه نفس هذا التنسيق , و لإلغاء النص المكتوب في المربعات يجب أن تقوم بتحديد كل مربع علي حده , فقم أولاً بإزالة التحديد من المربعات الأربعة معاً و ذلك بالنقر بزر الماوس في أي مكان فارغ في النموذج لإزالة التحديد من المربعات الأربعة , ثم قم بالنقر بزر الماوس علي المربع الأول لتحديده منفرداً , ثم أنقر نقراً



مزدوجاً بزر الماوس فوق خاصية Text في نافذة الخواص و أضغط مفتاح Delete في لوحة المفاتيح لإلغاء النص الموجود في هذا المربع , و بعد ذلك ستستخدم الأكواد كما سيتضح فيما بعد لكتابة رقم عشوائي في هذا المربع , و كرر نفس الشيء لإلغاء النصوص الموجودة في باقي المربعات .

أما بالنسبة للمربع الرابع و الموجود في السطر الثاني في النموذج و الذي عنوانه Lable4 قم بتحديد هذا المربع و اختار Text من نافذة الخواص و قم بتغييرها إلي Lucy Seven ثم قم بالنقر علي خاصية Font و قم بالنقر فوق الزر المستدير الموجود بجوار هذه الخاصية لفتح المربع الحواري الخاص بتغيير نوع الخط , في هذا المربع الحواري غير نوع الخط إلي Arial و تنسيق الخط إلي Bold و حجم الخط 18 و اضغط Ok و في قائمة الخواص قم بالنقر فوق خاصية Color و في السطر الخاص بهذه الخاصية قم بالنقر فوق المثلث الأسود الصغير الموجود بجوار أسم الخاصية لتظهر قائمة خيارات الألوان , من هذه الخيارات أختار Custom ثم قم بتغيير لون النصر إلي Dark Purple

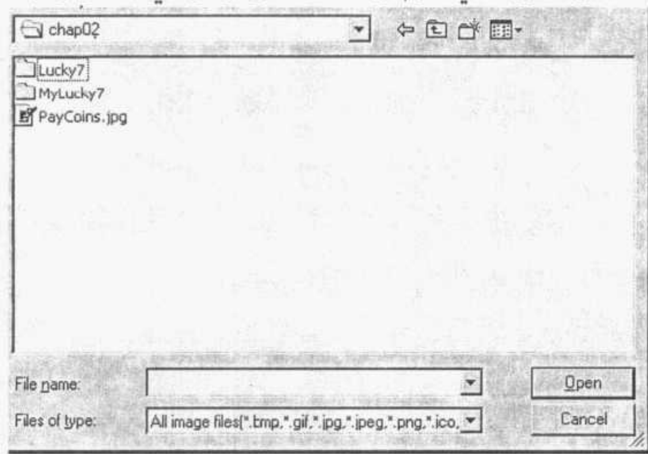
تغيير خواص مربع الصورة

في هذا المربع سوف تظهر صورة شخص يقوم بإعطاء النقود للفائز في هذه اللعبة , و سوف تستخدم صورة تحمل هذا المعني بامتداد JPEG و سوف تقوم بضبط حجم الصورة بنفس حجم المربع الذي ستوضع فيه هذه الصورة في النموذج .

و لتحديد خواص مربع الصورة , قم بالنقر فوق مربع الصورة لتحديده ,

ثم قم بالنقر فوق خاصية SizeMode في نافذة الخواص ثم قم بالنقر فوق السهم الأسود الصغير الموجود بجوار أسم هذه الخاصية في النافذة لتظهر قائمة بمجموعة الخيارات المتعلقة بهذه الخاصية , و من هذه الخواص أختار StretchImage و ذلك قبل فتح الصورة في المربع الخاص بالصورة ليقوم فيجوال أستوديو بضبط حجم الصورة لكي يلائم تماماً حجم المربع الذي ستوضع فيه هذه الصورة .

و لوضع الصورة في المربع الخاص بها في النموذج قم بالنقر فوق خاصية Image في نافذة الخواص نقراً مزدوجاً بزر الماوس ثم قم بالنقر فوق الزر المستدير الموجود بجوار هذه الخاصية لفتح المربع الحواري الخاص بتصفح الجهاز لاختيار الصورة التي تقوم بوضعها كما هو موضح في (شكل رقم ١-١٧)




المربع الحواري الخاص بوضع الصورة في المربع الخاص بها في النموذج
(شكل رقم ١-١٧)



في المربع الحواري الخاص بوضع الصورة في المربع الموضح في الشكل السابق قم بالتصفح للوصول إلي الصورة و أضغط زر Open ليقوم فيجوال بيسيك بفتح الصورة في المربع الخاص بها في النموذج .

سوف نقوم الآن بإخفاء الصورة عند بداية فتح البرنامج ثم نقوم بإظهار هذه الصورة بعد ذلك باستخدام الأكواد عندما يصل المستخدم إلي الرقم الرابح , حيث أنه من المفترض أن لا تظهر هذه الصورة عند فتح البرنامج و أنما تظهر فقط عند الوصول إلي الرقم الرابح .

و لإخفاء الصورة من المربع الموجودة فيه , أنقر فوق خاصية Visible في مجموعة Behavior في نافذة الخواص , ثم أختار False من قائمة الخيارات المعلقة بهذه الخاصية لإخفاء الصورة من النموذج عند بداية فتح البرنامج .

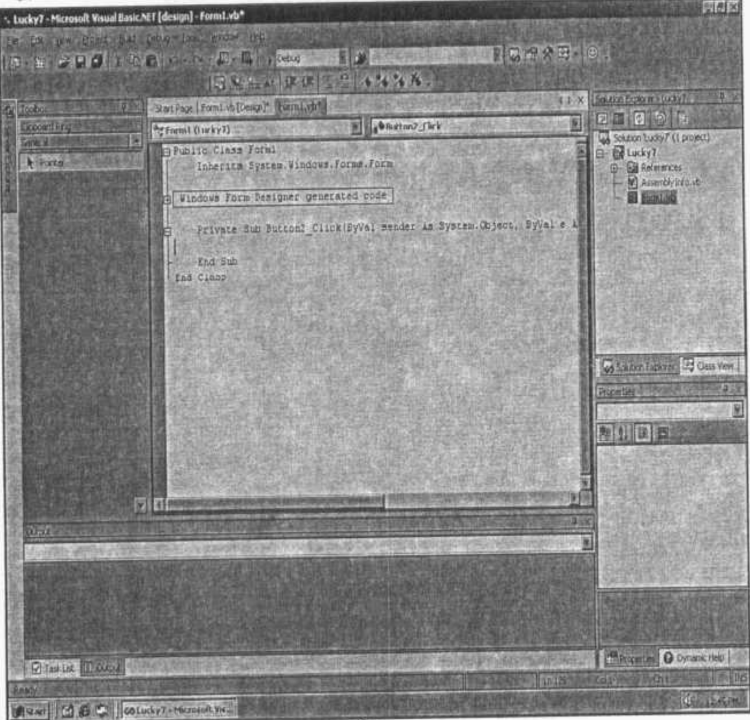
الخطوة الثالثة : كتابة الكود

عناصر التحكم التي قمت بإضافتها إلي النموذج مثل الأزرار و مربعات العناوين و الصورة عبارة عن كائنات برمجية , و بمجرد إضافة هذه الكائنات البرمجية إلي النموذج , و تحديد خواصها من نافذة الخواص تكون هذه الكائنات جاهزة للتفاعل مع المستخدم باستقبال الأحداث التي يقوم بها المستخدم مثل النقر علي الأزرار , و تشغيل هذه الأحداث من خلال البرنامج الذي سنقوم بكتابته و استخراج نتائج التشغيل , و الأكواد التي سنقوم بكتابتها الآن سوف تقوم باستخراج رقم عشوائي و يتم عرض هذا الرقم في مربعات العناوين و استخراج الرقم الفائز من بين هذه الأرقام و خروج الصورة التي تعتبر رسالة تهنئة بالفوز عند اختيار الرقم الرابح .

استخدام محرر الأكواد

لفتح محرر الأكواد أنقر نقراً مزدوجاً فوق زر End و هو الزر الثاني في النموذج ليتم بذلك فتح محرر الأكواد في وسط واجهة البرنامج , كما يوضحه (شكل رقم ١-١٨) , و يحتوي محرر الأكواد الذي يتم فتحه بالنقر نقراً مزدوجاً علي زر End في النموذج علي الأكواد التي قام فيجوال أستوديو بكتابتها لإنشاء هذا النموذج و أدوات التحكم التي يحتوي عليها , و في محرر الأكواد تم تجميع الجمل البرمجية المستخدمة في تشغيل عناصر التحكم في النموذج تم تجميعها في مجموعات تسمي أجراء Procedure و النوع الشائع من هذا الأجراء هو الأجراء الفرعي Subruotine أو Sub Procedures و يحتوي علي الكلمة المفتاحية Sub في السطر الأول و في السطر الأخير من هذه المجموعة من الجمل حيث يبدأ بكلمة Sub في أول سطر في مجموعة الجمل و ينتهي بكلمة End Sub في السطر الأخير من مجموعة الجمل , و يتم تنفيذ هذه المجموعة من الجمل البرمجية الخاصة بتشغيل النموذج عند حدوث حدث معين Event مثل الضغط علي أحد أزرار النموذج حيث يتم ربط هذه المجموعة من الجمل بحدوث هذا الحدث و هو الضغط علي زر معين , و تسمي عملية الربط بين الحدث و بين تشغيل مجموعة الجمل البرمجية Event Handler





محرر الأكواد

(شكل رقم ١-١٨)

و بمجرد النقر نقراً مزدوجاً علي زر End في النموذج يقوم فيجوال أستوديو أوتوماتيكياً بإضافة السطر الأول و السطر الأخير من مجموعة الجمل الخاصة بتشغيل هذا الزر في محرر الأكواد كما يوضحها الكود التالي , و يمكن التعرف علي هذه المجموعة من الجمل البرمجية الخاصة بالنموذج من خلال عبارة Windows Form Designer Generated Code الموجودة في محرر الأكواد

```
• Private Sub Button2_Click(ByVal sender As System.Object, _
• ByVal e As System.EventArgs) Handles Button2.Click
• End Sub
```

و الكود السابق يوضح السطر الأول و الأخير من مجموعة الأكواد التي تمثل مجموعة الجمل البرمجية Procedure الخاصة بتشغيل زر النموذج , و يتم وضع الأكواد الخاصة بتحديد ماذا يحدث عند الضغط علي هذا الزر بين هذين السطرين , و يكون الحدث Event المتعلق بتشغيل هذه المجموعة من الأكواد هو الضغط علي زر End

و نريد أن ينتهي عمل البرنامج بالضغط علي زر End و لذلك قم بكتابة كلمة End بين السطرين الموضحين في الكود السابق في محرر الأكواد ثم قم بالضغط علي مفتاح السهم لأسفل في لوحة المفاتيح لتخبر فيجوال أستوديو أن يقوم بإغلاق البرنامج عند الضغط علي زر End في النموذج .

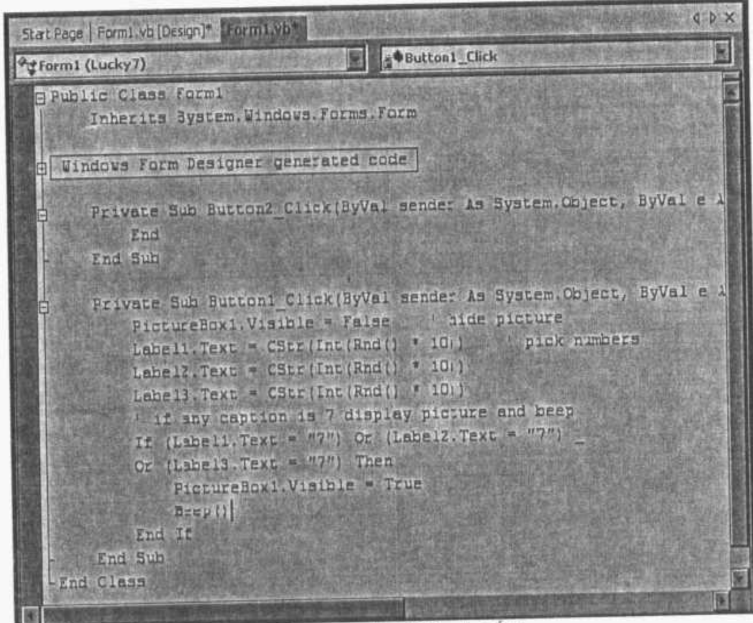
الكود الخاص بزر Spin

لإظهار النموذج مرة أخري قم بعرض نمط التصميم في واجهة البرنامج و ذلك بالنقر فوق الزر الخاص بنمط التصميم في نافذة المتصفح في يسار البرنامج أو بالنقر علي زر Form1.vb في أعلي واجهة البرنامج .

بعد إظهار النموذج في نمط التصميم أنقر نقراً مزدوجاً فوق زر Spin لفتح الأكواد الخاصة بهذا الزر في محرر الأكواد , و كما حدث مع زر End يحدث مع زر Spin و يقوم البرنامج بعرض السطر الأول و السطر الأخير من مجموعة الجمل المتعلقة بتشغيل هذا الزر , و بين هذا السطرين قم بكتابة مجموعة الأسطر التالية :

- PictureBox1.Visible = False ' hide picture
- Label1.Text = CStr(Int(Rnd() * 10)) ' pick numbers
- Label2.Text = CStr(Int(Rnd() * 10))
- Label3.Text = CStr(Int(Rnd() * 10))
- if any caption is 7 display picture and beep
- If (Label1.Text = "7") Or (Label2.Text = "7") _
- Or (Label3.Text = "7") Then
- PictureBox1.Visible = True
- Beep()
- End If

بحيث بعد الانتهاء من كتابة الأكواد السابقة تكون الأكواد الخاصة بزر Spin في محرر الأكواد كما يوضح (شكل رقم ١-١٩) .



الأكواد الخاصة بزر Spin
(شكل رقم ١-١٩)

و بالنظر إلي الكود السابق الخاص بتشغيل الزر Spin تجد أنه بقيام المستخدم بالنقر علي هذا الزر بزر الماوس يقوم هذا الزر بأداء ثلاثة وظائف أساسية أول هذه الوظائف هي أخفاء الصورة و الوظيفة الثانية هي أنشاء ثلاثة أرقام عشوائية تظهر في مربعات العناوين الخاصة بالأرقام الموجودة في النموذج و الوظيفة الثالثة هي إظهار الصورة عند الوصول إلي الرقم الفائز و هو رقم ٧ .

و يتم أداء الوظيفة الأولي و هي أخفاء الصورة باستخدام السطر التالي من الكود .

- PictureBox1.Visible = False

و في الكود السابق نجعل قيمة خاصية Visible المسئولة عن إظهار الصورة الموجودة في المربع PictureBoxe1 نجعل قيمتها False مما يؤدي إلي أخفاء الصورة .

أما الثلاثة أسطر البرمجية التالية لهذا السطر فهي خاصة بأداء الوظيفة الثانية للزر وهي إظهار الأرقام العشوائية , فكل سطر من الأسطر الثلاثة يقوم بإظهار رقم عشوائي باستخدام دالة Rand الخاصة بإنشاء الأرقام العشوائية التي تقع قيمها بين الصفر و رقم ٩ كما في الكود التالي :

- abel1.Text = CStr(Int(Rnd() * 10))

و وضع الرقم العشوائي في مربع العنوان Lable1 و الكود التالي له يقوم بإنشاء رقم عشوائي و وضعه في مربع العنوان Lable2 و الكود التالي له يقوم بإنشاء رقم عشوائي و وضعه في مربع العنوان Lable3



و بعد اختيار الأرقام العشوائية لوضع هذه الأرقام العشوائية في مربعات العناوين الخاصة بها يجب أولاً تحويل هذه الأرقام إلي متغيرات نصية String قبل وضعها في المربعات الخاصة بها في النموذج , و قد استخدمنا في تحويل نوع البيانات من بيانات عددية صحيحة إلي متغيرات نصية استخدمنا دالة CStr في نفس الكود الخاص باختيار الرقم العشوائي الذي يتم وضعه في كل مربع من المربعات الثلاثة .

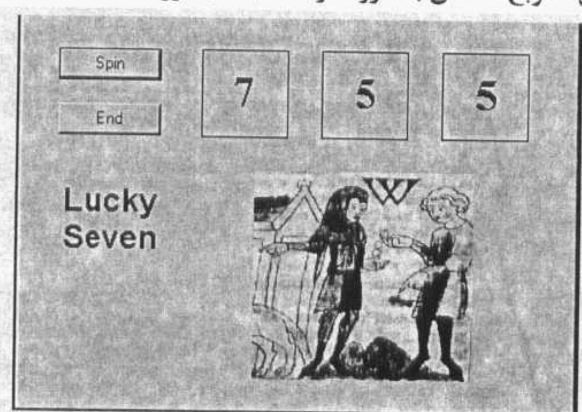
و باقي الأسطر الموجودة في الجود تستخدم جملة If الشرطية في أظهار الصورة في حالة خروج الرقم 7 من بين الأرقام العشوائية التي يتم اختيارها من بين الأرقام من واحد إلي ٩ كما يلي :

```
' if any caption is 7 display picture and beep
If (Label1.Text = "7") Or (Label2.Text = "7") _
Or (Label3.Text = "7") Then
PictureBox1.Visible = True
Beep()
End If
```

تجربة البرنامج

لتجربة البرنامج الذي قمت بإنشائه في برنامج فيجوال أستوديو قم بفتح النموذج الخاص باللعبة التي قمت بإنشائها في البرنامج و أضغط مفتاح F5 في لوحة المفاتيح , و في هذه الحالة أما أن يتم تشغيل البرنامج و أما أن تظهر رسالة خطأ في حالة ما إذا كان هناك خطأ ما في كتابة البرنامج , و في حالة ظهور رسالة خطأ يجب عليك تصحيح هذا الخطأ أولاً بناء علي التعليمات التي تظهر في رسالة الخطأ قبل أن تتمكن من تشغيل البرنامج .

و في حالة عدم ظهور رسالة خطأ أثناء تجربة البرنامج , أو في حالة قيامك بتصحيح الخطأ في حال ظهوره بالضغط علي مفتاح F5 يقوم فيجوال أستوديو بعرض النموذج في شكله النهائي كما هو موضح في (شكل رقم ١- ٢٠) و بالضغط علي مفتاح Spin في النموذج يتم عرض ثلاثة أرقام عشوائية كما يوضحها الشكل , و في حالة ظهور الرقم ٧ ضمن الأرقام العشوائية التي تظهر معه الصورة في المربع الخاص بالصورة كرسالة تهنئة بالفوز .



البرنامج في شكله النهائي
(شكل رقم ١-٢٠)

أخراج البرنامج في صورته النهائية

سوف تقوم الآن بإخراج البرنامج في صورة قابلة للاستخدام و التشغيل بامتداد exe. ليعمل في نظام تشغيل الويندوز , و تتمثل عملية أخراج المشروع في صورته النهائية في قيام مفسر الأكواد ompiler بتفسير الأكواد

الخاصة بكتابة المشروع و ترجمة هذه الأكواد في شكل برنامج قابل للاستخدام .

و لإخراج المشروع في شكله النهائي أختار Configuration Manager من قائمة Build ليظهر بذلك المربع الحواري الخاص بإعداد البرنامج كما يوضحه (شكل رقم ١-٢١) , و الذي يمنحك الاختيار بين أخراج Release Builds البرنامج أو تصحصح بما به من أخطاء Debug Builds

Active Solution Configuration:
Release

Project Contexts (check the project configurations to build or deploy):

Project	Configuration	Platform	Build
Lucky7	Release	.NET	☑

Close Help

المربع الحواري الخاص بإخراج البرنامج
(شكل رقم ١-٢١)

في المربع الحواري الخاص بإخراج البرنامج الموضح في الشكل السابق أختار البرنامج من قائمة البرامج التي تظهر في المربع الحواري و أختار Release من قائمة Active Solution Configuration الموجودة بأعلى المربع

الحواري و أضغط زر Ok ليتم بذلك تفسير الأكواد الموجودة في البرنامج .

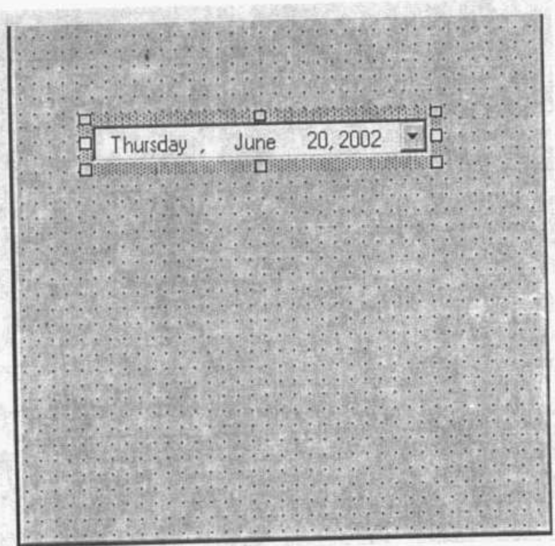
من قائمة Build أختار Build Solution ليقوم فيجوال أستوديو بإنشاء ملف في المسار المحدد في المربع الحواري الخاص بإنشاء المشروع , و وضع البرنامج النهائي MyLucky7.exe في هذا الملف , ويمكنك الحصول علي هذا الملف بتصفح الجهاز للوصول إليه و نسخ البرنامج النهائي إلي المكان الذي تريده في الجهاز .

المشروع الثالث

برنامج تاريخ الميلاد

لإغلاق المشروع الحالي المفتوح في البرنامج لإعداد البرنامج لإنشاء مشروع أخري أختار Close Solution من قائمة File لإغلاق المشروع الحالي المفتوح في البرنامج قبل البدء في العمل في مشروع جديد , و للبدء في مشروع جديد اختار New من قائمة File ليظهر بذلك المربع الحواري الخاص بإنشاء ملف جديد , من هذا المربع الحواري اختار Windows Application و قم بكتابة أسم المشروع الجديد و اختيار المسار الذي يقوم فيه البرنامج بوضع الملفات الخاصة بهذا المشروع الجديد و أضغط زر Ok ليقوم فيجوال أستوديو بإنشاء نموذج فارغ , و في شريط الأدوات Toolbox الموجود في يسار واجهة البرنامج قم بالنقر فوق DateTimePicker ثم قم برسم مربع للوقت و التاريخ في أعلي النموذج كالموضح في (شكل رقم ١-٢٢) .





المربع الخاص بالوقت و التاريخ

(شكل رقم ١-٣٢)

و بشكل افتراضي يظهر الوقت و التاريخ الحالي في المربع الخاص بالوقت و التاريخ الذي تقوم بإنشائه , و يمكنك تغيير هذا الوقت و التاريخ بعد ذلك عن طريق القيمة التي يتم إعطاؤها لكائن الوقت و التاريخ .

ثم قم بالنقر فوق Button في شريط الأدوات Toolbox ثم قم برسم زر بأسفل مربع الوقت و التاريخ , و الآن سوف تقوم بإضافة وظيفة Procedure لهذا الزر , هذه الوظيفة سيتم تشغيلها بمجرد حدوث الحدث Event المرتبط بتشغيل الزر و هو النقر فوق الزر بزر الماوس , و ذلك عن طريق كتابة الأسطر البرمجية الخاصة بهذه الوظيفة في محرر الأكواد , و لإظهار محرر الأكواد

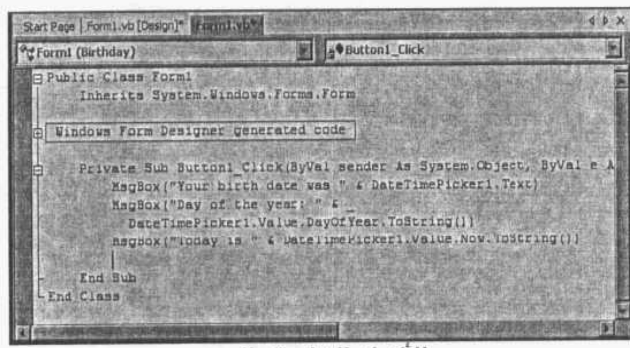
ليعرض الأكواد المتعلقة بالزر أنقر نقراً مزدوجاً فوق الزر الذي قمت بإضافته في النموذج , ليظهر بذلك محرر الأكواد و الذي يعرض كما سبق و أن أوضحنا الجمل المتعلقة بتشغيل الزر , بين هذه الجمل أكتب السطر التالي , و الخاص بتحديد الحدث الخاص بتشغيل الزر و هو النقر فوق الزر بزر الماوس , و ما يحدث نتيجة حدوث هذا الحدث .

```
• MsgBox("Your birth date was " & DateTimePicker1.Text)
• MsgBox("Day of the year: " & _
• DateTimePicker1.Value.DayOfYear.ToString())
• MsgBox("Today is " & DateTimePicker1.Value.Now.ToString())
```

الأكواد السابقة تقوم بإظهار ثلاثة مربعات حوار , تحتوي هذه المربعات علي معلومات عن الوقت و التاريخ , السطر الأول من الجمل السابقة خاص بإظهار النص المتعلق بعرض الوقت و التاريخ , و السطر الثاني خاص بفتح المربع الحواري MesgBox و الخاص بعرض قيمة المتغير اللفظي التي تساوي تاريخ الميلاد , و السطر الثالث و الرابع تستخدم في أجراء عملية حسابية متعلقة باحتساب تاريخ الميلاد , و أظهار المربع الحواري , و تحويل تاريخ الميلاد إلي متغير لفظي , و السطر الخامس يقوم بفحص الوقت و التاريخ الخاص بجهاز الكمبيوتر و يقوم بتحويلها إلي متغير لفظي .

و بعد كتابة الأكواد السابقة في محرر الأكواد بين الأكواد الخاصة بالزر يجب أن تكون الأكواد الخاصة بالزر كما يوضح (شكل رقم ١-٢٣) .





الأكواد الخاصة بالزر

(شكل رقم ١-٢٣)

أضغط Save All لحفظ الأكواد .

اختبار البرنامج

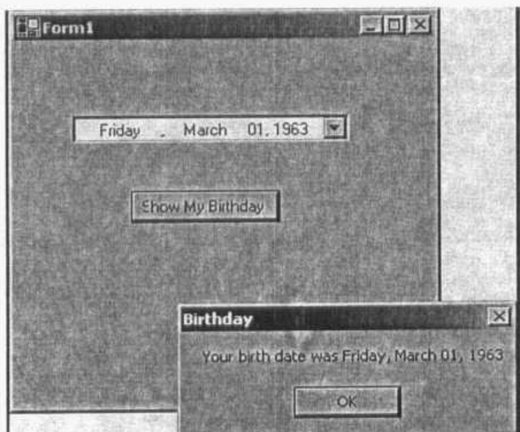
لاختبار البرنامج أضغط زر Start الموجود في شريط الأدوات , ثم قم في النموذج بفتح القائمة المنسدلة التي تعرض التاريخ بالنقر علي السهم الأسود الصغير الموجود في اليمين كما يوضح (شكل رقم ١-٢٤) , ثم قم باختيار تاريخ ميلادك



اختبار البرنامج

(شكل رقم ١-٢٤)

ثم قم بالضغط فوق الزر لتظهر رسالة مكتوب بها تاريخ ميلادك كالموضحة في (شكل رقم ١-٢٥) .



عرض تاريخ الميلاد

(شكل رقم ١-٢٥)





واجهة برنامج فيجوال أستوديو

واجهة برنامج فيجوال أستوديو

تتشابه Visual Basic.Net كثيراً مع لغة C# من حيث قوة كل منهما و استخدام قواعد متشابهة في كتابة الأكواد , غير أن لغة C# تستخدم الأقواس و الفاصلة المنقوطة كثيراً حيث في لغة C# لابد و أن ينتهي كل سطر برمجي بالفاصلة المنقوطة , و كلتا اللغتين يمكنها العمل في بيئة دوت نت , كما يمكن استخدام اللغتين معاً في مشروع واحد , و لعل الفارق الوحيد بين لغة Visual Basic.Net و بين لغة C# هو أن لغة Visual Basic.Net أسهل في التعلم و أسهل في الاستخدام .

و ترتبط لغة Visual Basic.Net بمعظم منتجات شركة مايكروسوفت فلغة ASP تستخدم Visual Basic.Net في عمل صفحات ويب ديناميكية , و حزمة برامج Microsoft Office مثل Word و Excel و PowerPoint و غيرها تستخدم Visual Basic.Net كما أن تعلم لغة Visual Basic.Net يسهل عليك عملية تعلم لغة ASP و لغة VBA و لتشابه لغة Visual Basic.Net مع العديد من لغات البرمجة , و لسهولة تعلم هذه اللغة فإن تعلم هذه اللغة يعتبر الطريقة السهلة أو المدخل لتعلم العديد من لغات البرمجة .

دوت نت

بيئة دوت نت Net. عبارة عن مجموعة كبيرة من مكتبات العديد من لغات البرمجة مثل Visual Basic و غيرها , و تحتوي هذه المجموعة من المكتبات علي الأكواد و الفئات التي يدونها كان يجب علي المبرمج كتابتها سطراً



بسطر مثل أكواد و فئات التعامل مع الملفات و التعامل مع قواعد البيانات و التعامل مع النماذج , و ترتبط Visual Basic بمكتبات Net. لدرجة أنها لا يمكن أن تعمل يدونها , فالعديد من الأدوات التي تستخدمها Visual Basic مصدرها مكتبات Net. فالاثنين مرتبطين بعضهما ببعض ارتباطا كبيراً لدرجه أنه لا يمكن التمييز أي الخواص مرتبط بـ Visual Basic و أيها مرتبط بمكتبات Net. و قامت شركة مايكروسوفت بإصدار الإصدار الثالث من مكتبات Net. و الذي يعرف باسم WinFX و هذا الإصدار من مكتبات Net. تم تصميمه بحيث يتعامل مع نظام تشغيل ويندوز فيستا و قد تم إصداره مع ويندوز فيستا في نفس الوقت , و يستخدم ويندوز فيستا أدوات و نوافذ هذا الإصدار من مكتبات Net. و كذلك خلفية ويندوز فيستا الشفافة و قد تم تصميم هذا الإصدار من مكتبات Net. ليناسب احتياجات مصممي الجرافيك .

و يقوم هذا الكتاب بشرح Visual Basic.Net من البداية و لذلك لا يتطلب منك لقراءة هذا الكتاب أن يكون لديك أي خلفية مسبقة بأي من لغات البرمجة كما يتدرج هذا الكتاب ليناسب احتياجات المبرمج المتوسط و المبرمج المحترف , حيث يتناول هذا الكتاب أيضاً العديد من الأسرار و الخفايا المتعلقة بـ Visual Basic.Net و العديد من التفاصيل المتعلقة بها

بيئة فيجوال بيسيك دوت نت IDE

تحتوي بيئة فيجوال بيسيك دوت نت Integrated Development

Environment (IDE) علي ألاف الأوامر و القوائم و أشرطة الأدوات و النوافذ و القوائم المختصرة المتعلقة بتشغيل مشروعات فيجوال بيسيك أو تعديل هذه المشروعات أو تنقية هذه المشروعات من الأخطاء , و يمكنك تعديل واجهة Visual Studio IDE علي حسب رغبتك و وفقاً لما يتناسب مع المشروع الذي تقوم به , فيمكنك أخفاء و إظهار القوائم و أشرطة الأدوات و النوافذ أو نقلها أو تعديلها , كما يمكنك أيضاً أن تقوم بعمل مجموعات خاصة بك من أشرطة الأدوات , و يمكنك إعادة ترتيب الأدوات في أشرطة الأدوات , كما يمكنك أيضاً التحكم في خواص محرر الأوامر فيمكنك تغيير ألوان الأنواع المختلفة من النصوص , و يمكنك التحكم في الطريقة التي تترك بها الفراغات بين أجزاء النص , كل هذه الخواص تمكنك من الوصول إلي بيئة العمل التي تناسب احتياجاتك و ذوقك فيمكنك إخفاء بعض الأدوات التي لست في حاجة إليها في وقت معين ثم إظهارها مرة أخري عند الحاجة إليها , و يمكنك نقل أجزاء مساحة العمل من مكان إلي أخر علي حسب احتياجاتك , كما يمكنك وضع كل الأدوات التي تستخدمها في الوقت الحالي في شريط أدوات واحد و تقوم بوضع هذا الشريط في المكان الذي تريده في مساحة العمل , و بنفس الطريقة يمكنك عمل أكثر من شريط أدوات للتعامل مع الأكواد و النماذج و قواعد البيانات .

و بهذا تتسم بيئة فيجوال بيسيك دوت نت IDE بالمرونة و لهذا السبب من



الممكن أن تختلف البيئة الموجودة في الصور الموجودة في الكتاب عن البيئة التي تعمل بها و الموجودة علي جهازك , و من الممكن أن تختلف الأوامر الموجودة في الصور عن الأوامر الموجودة علي جهازك , و من الممكن أن توجد هذه الأوامر و لكن في أماكن أخري , و لتجنب ذلك لا تقوم بإجراء أي تعديل علي مكونات بيئة فيجوال بيسيك دوت نت الموجودة لديك إلا بناء علي التعليمات الخاصة بذلك و الموجودة في الكتاب حتى تتشابه بيئة دوت نت الموجودة لديك مع ما تراه في الكتاب و يكون من السهل الوصول إلي مكونات هذه البيئة حتى يسهل عليك متابعة الأمثلة المعروضة في الكتاب و تنفيذها بيدك .

المشروعات

المشروع Project هو عبارة عن مجموعة من الملفات التي ينتج عنها مجموعة من المخرجات , هذه المخرجات قد تكون برامج قابلة للتنفيذ أو قد تكون مكتبات روابط ديناميكية Dynamic-link Library (DLL) أو فئات Classes يمكن استخدامها في مشروعات أخري , أو أدوات تحكم يمكن استخدامها في نماذج أخري .

و مجموعة المشروعات التي يمكن التعامل معها في وقت واحد تسمي Solution و للإيضاح , أفترض أن أحدي الشركات تريد عمل مشروع للتعامل مع قواعد البيانات , و تريد عمل مشروعات أخري لكل مندوب مبيعات من مندوبي الشركة لتعامله مع قواعد البيانات , فهذه المشروعات بما أنها يجب أن يتم

التعامل معها مرة واحدة لارتباطها بعضها البعض و يجب استخدامها معاً و الوصول إليها معاً في وقت واحد , فإن هذه المشروعات تحتوي علي ملفات مشتركه فيما بينها , فهذه المشروعات كلها تعتبر مجموعة واحدة تسمي Solution

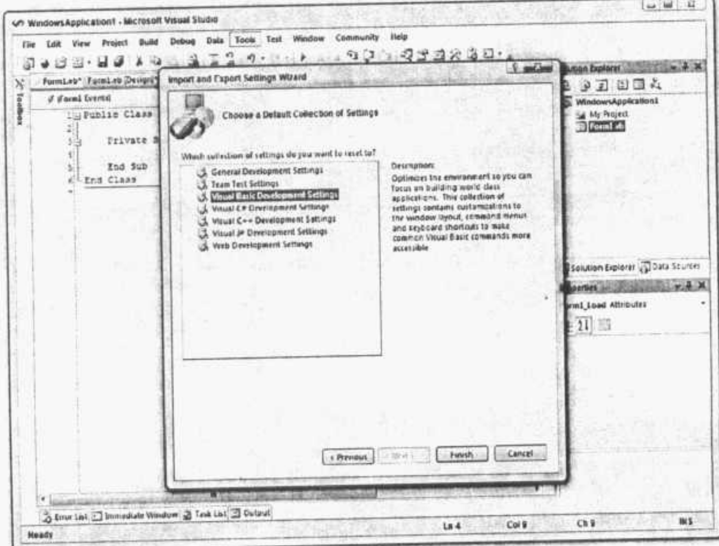
فيجوال أستوديو

عندما تقوم بتشغيل برنامج فيجوال أستوديو لأول مرة تظهر نافذة البداية الخاصة بهذا البرنامج بشكل افتراضي , و لتغيير ما يقوم به البرنامج عند بداية تشغيله , أختار أمر Command الموجود في قائمة Tools ثم قم بتحديد مربع الخيار Show all settings ليقوم البرنامج بعرض كل الخيارات المتعلقة ببداية التشغيل, من هذه الخيارات يمكنك اختيار ما يقوم به البرنامج عند بداية تشغيله .

و يختلف شكل واجهة البرنامج باختلاف العملية التي يقوم بها البرنامج , و يختلف أيضاً باختلاف إصدار الويندوز الذي يعمل به البرنامج ,و في حالة تشغيل البرنامج في بيئة ويندوز فيستا يختلف شكل واجهة البرنامج باختلاف الأستايل الذي اخترته من بين الأستايلات الموجودة في ويندوز فيستا , و يختلف شكل البرنامج أيضاً في ويندوز فستا باختلاف نوع حساب المستخدم الذي قمت بفتح الجهاز به , و لهذا السبب قد تختلف الصور الموجودة في الكتاب عن الصور التي تقوم بالتطبيق عليها علي جهازك , كما يختلف شكل واجهة البرنامج أيضاً باختلاف الإصدار من البرنامج الذي قد تعمل عليه , و قد تم استخدام إصدار ٢٠٠٥ في الأمثلة المشروحة في هذا الكتاب , كما قد يختلف شكل البرنامج

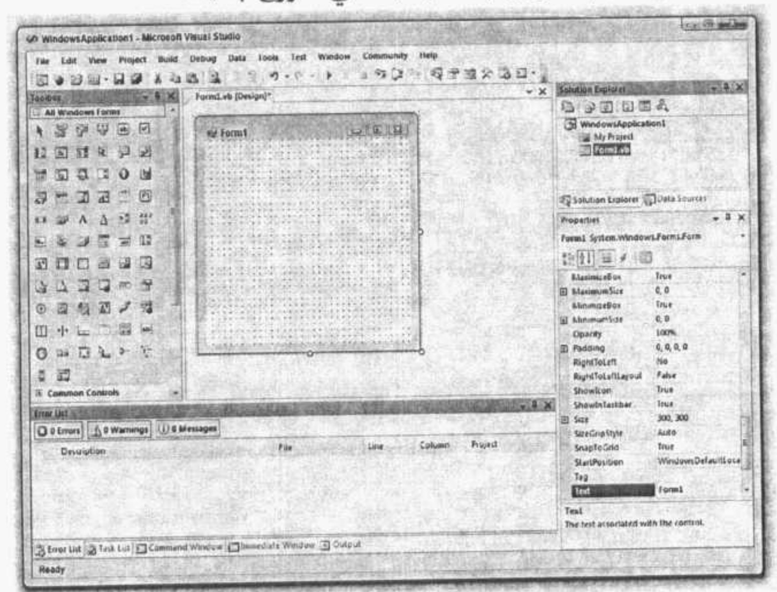
أيضاً باختلاف الأعدادات التي تقوم باختيارها عند تنصيب البرنامج كاختيارك C# عند بداية تنصيب البرنامج أو اختيارك C++ أو اختيارك eb Development Settings

و يمكنك تغيير هذه الأعدادات بعد تنصيب البرنامج باختيار Import and Export Settings من قائمة Tools ليظهر بذلك معالج الاستيراد و التصدير كما يوضحه (شكل رقم ١-١) من أخر نافذة من المعالج اختار Reset all Settings ثم أضغط زر Next وفي الصفحة التالية أختار Save current settings لحفظ الأعدادات الحالية ثم اضغط زر Next وفي الصفحة الأخيرة من المعالج كما يوضحها الشكل أختار نوع الأعدادات التي تريدها ثم أضغط زر Finish



نافذة اختيار أعدادات Visual Studio
(شكل رقم ١-١)

و بعد إعادة تشغيل البرنامج يمكنك اختيار New Project من قائمة File كما يوضحها (شكل رقم ١-٢) للبدء في مشروع جديد .



نافذة البدء في مشروع جديد
(شكل رقم ١-٢)





قوائم برنامج
Visual Studio

- قائمة File
- قائمة Edit
- قائمة View
- قائمة Project
- قائمة Build
- قائمة Debug
- قائمة Data
- قائمة Format
- قائمة Tools
- قائمة Test
- قائمة Windows
- قائمة Community

تعلم بنفسك Visual Basic.net 2005

قائمة File

تحتوي قوائم برنامج Visual Studio علي مجموعة كبيرة من الأوامر الخاصة بالتعامل مع المشروعات المفتوحة في البرنامج , و تتشابه هذه القوائم في محتواها و ترتيبها مع القوائم الموجودة في أي من تطبيقات الويندوز , مثل قائمة File و قائمة View و قائمة Help و ما إلي ذلك مع اختلاف الأوامر التفصيلية التي تحتوي عليها هذه القائمة و الخاصة ببيئة فيجوال بيسيك دوت نت , و كما في حالة الكثير من تطبيقات الويندوز يمكنك تغيير القوائم في بيئة فيجوال بيسيك , فيمكنك إضافة أو إزالة أوامر كما يمكنك إعادة ترتيب القوائم كما يمكنك أيضاً إضافة أو إزالة أوامر من هذه القوائم , و لكن لا ينصح بتغيير الأوامر الموجودة في القوائم و تركها علي وضعها الافتراضي , و ينصح بعمل قائمة إضافية لكي تضيف بها ما تشاء من أوامر مع عدم المساس بالقوائم الافتراضية الموجودة في البرنامج .

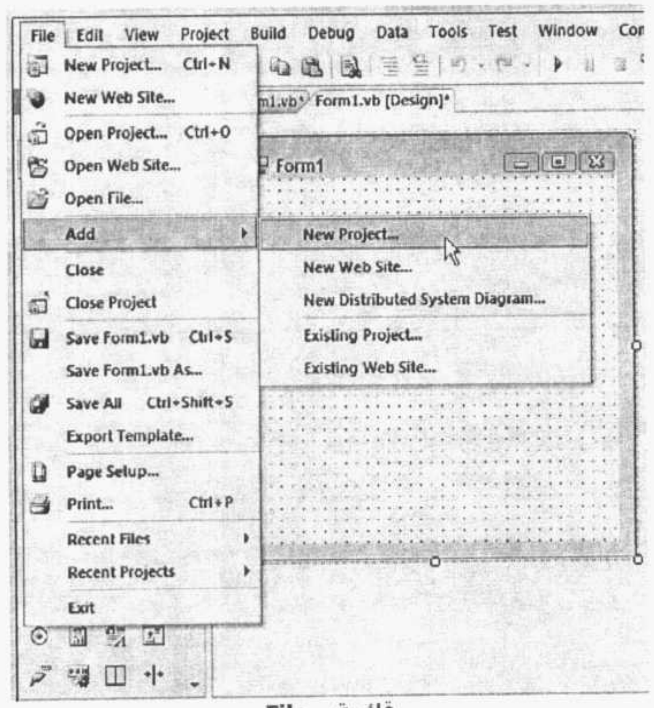
و يوجد لمعظم الأوامر التي تحتوي عليها القوائم اختصارات لوحة مفاتيح لتنفيذ هذه الأوامر بمجرد الضغط علي هذه المفاتيح في لوحة المفاتيح بدلاً من اختيار هذه الأوامر من القوائم الموجودة بها , مثل اختصار Ctrl+N لفتح المربع الحواري الخاص بإنشاء مشروع جديد , و يمكنك أن تتعرف علي الاختصارات الخاصة بالأوامر في حالة وجودها بمجرد النظر في القائمة التي تحتوي علي هذه الأوامر , حيث تحتوي هذه القوائم علي الاختصار الخاص بكل أمر مكتوب بجوار الأمر في القائمة مثل معظم البرامج الأخرى , كما يمكن



تنفيذ الكثير من الأوامر بمجرد النقر علي الأداة الخاصة بتنفيذ هذا الأمر في شريط الأدوات , فالكثير من الأوامر لها أدوات عبارة عن أيقونات صغيرة في شريط الأدوات يمكن تنفيذ هذه الأوامر بمجرد النقر علي هذه الأيقونات الخاصة بهذه الأوامر .

و هناك الكثير من الأوامر الخاصة بنقاط معينه في المشروع الذي تقوم بتنفيذه هذه الأوامر موجودة في القوائم المختصرة التيّ تظهر بالنقر بزر الماوس الأيمن في أماكن معينة في البرنامج , كما يعرض فيجوال أستوديو قوائم مختلفة تحتوي علي أوامر مختلفة علي حسب المحرر المفتوح في البرنامج , فعلي سبيل المثال عندما تقوم بفتح محرر الأكواد Code Editor تختفي قائمة Format حيث لا يكون لها عمل مع هذا المحرر .

قائمة File كما يوضحها (شكل رقم ١-٣) تحتوي علي مجموعة الأوامر الخاصة بإنشاء و فتح و حفظ و إغلاق المشروعات و الملفات المتعلقة بهذه المشروعات .



قائمة File

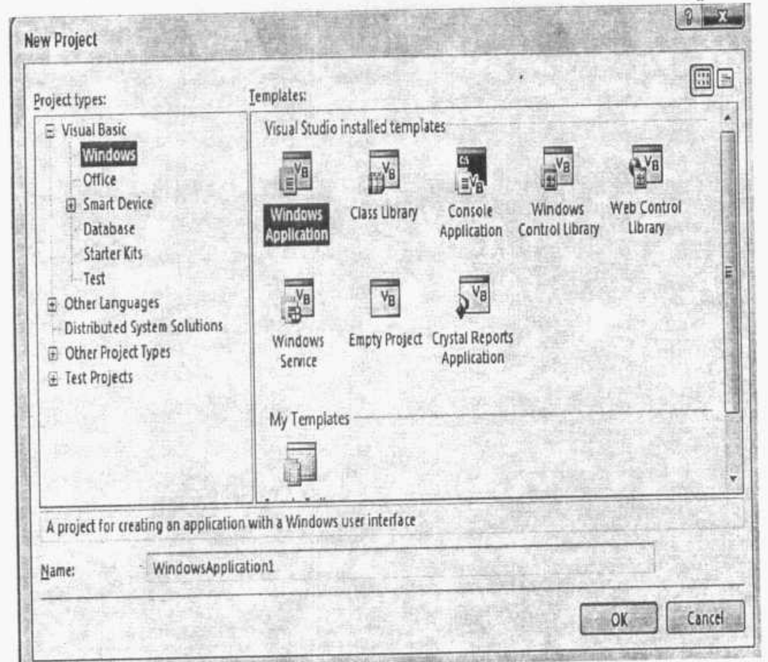
(شكل رقم ١-٣)

أمر New Project

تبدأ قائمة File كما يوضحها الشكل السابق بأمر New Project و الخاص بإنشاء مشروع جديد , و باختيار هذا الأمر من القائمة أو الضغط علي اختصار لوحة المفاتيح الخاص بهذا الأمر كما هو مكتوب بجوار الأمر في القائمة يظهر المربع الحواري الخاص بإنشاء مشروع جديد كما هو موضح



في (شكل رقم ١-٤) .



المربع الحواري الخاص بإنشاء مشروع جديد

(شكل رقم ١-٤)

و بإظهار المربع الحواري الخاص بإنشاء مشروع جديد كما يوضحه الشكل السابق بتم إغلاق المشروع الحالي المفتوح في البرنامج , و يتيح المربع الحواري الخاص بإنشاء مشروع جديد الموضح في الشكل السابق أنشاء العديد من أنواع المشروعات الجديدة .



أمر New Web Site

باختيار أمر New Web Site من قائمة File يتم إغلاق المشروع الحالي المفتوح في البرنامج و يبدأ البرنامج في فتح مشروع صفحة ويب جديد و يظهر المربع الحواري الخاص باختيار نوع صفحة الويب التي تريد إنشاؤها مثل صفحة ASP.Net

أمر Open Project

باختيار هذا الأمر من قائمة File يتم إغلاق المشروع الحالي المفتوح في البرنامج و يمكنك فتح مشروع قائم موجود مسبقاً .

أمر Open Web Site

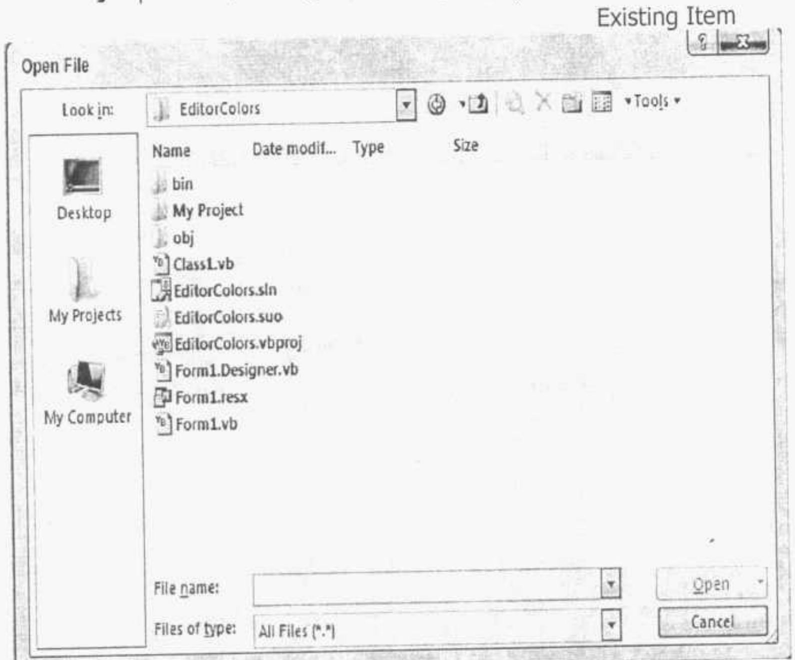
باختيار هذا الأمر من قائمة File يمكنك فتح مشروع موقع ويب موجود مسبقاً و إغلاق المشروع الحالي المفتوح في البرنامج .

أمر Open File

باختيار هذا الأمر من قائمة File يتم فتح المربع الحواري الموضح في (شكل رقم ١-٥) , و الخاص باختيار الملف الذي تريد فتحه في البرنامج , و تحتوي بيئة فيجوال بيسيك دوت نت علي محررات يمكنك استخدامها في تعديل الملف الذي تقوم بفتحه , فعلي سبيل المثال محرر الصور Bitmap Editor يمكنك استخدامه في تعديل الصور التي تقوم بفتحها في البرنامج من حيث تحديد حجم هذه الصور أو تغيير عدد الألوان التي تحتوي عليها الصور التي تقوم بفتحها في البرنامج كما يمكنك أيضاً بواسطة هذا المحرر الرسم في هذه الصور , و عندما تقوم بإغلاق الملف الذي



قمت بفتحه يسألك فيجوال أستوديو ما إذا كنت تريد حفظ التغييرات التي قمت بإجرائها علي الملف الذي قمت بفتحه في البرنامج أم لا و يتيح لك أيضاً اختيار المكان الذي تريد حفظ الملف فيه بعد تعديله , و لا يقوم فيجوال أستوديو بإضافة الملف الذي قمت بفتحه في البرنامج إلي مشروعك بشكل تلقائي و لكن يمكنك فعل ذلك باستخدام أمر Add Existing Item



المربع الحواري الخاص بفتح ملف

(شكل رقم ١-٥)

القائمة الفرعية Add

تتيح هذه القائمة الفرعية من قائمة File إضافة عناصر جديدة , و تفيد الأوامر الموجودة في هذه القائمة في حالة ما إذا كنت تعمل في مشروعين مترابطين في البرنامج حيث من خلال هذه القائمة يمكنك إضافة مشروع إلي المشروع المفتوح حالياً في البرنامج .

أمر Close

يستخدم أمر Close في إغلاق المحرر المفتوح في البرنامج .

أمر Close Project

يستخدم هذا الأمر في إغلاق كل المشروع المفتوح حالياً في البرنامج و إغلاق كل الملفات المتعلقة بهذا المشروع .

أمر Save Form1.vb

يستخدم هذا الأمر في حفظ الملف التعديلات التي أجريت علي الملف الحالي المفتوح في البرنامج .

أمر Save Form1.vb As

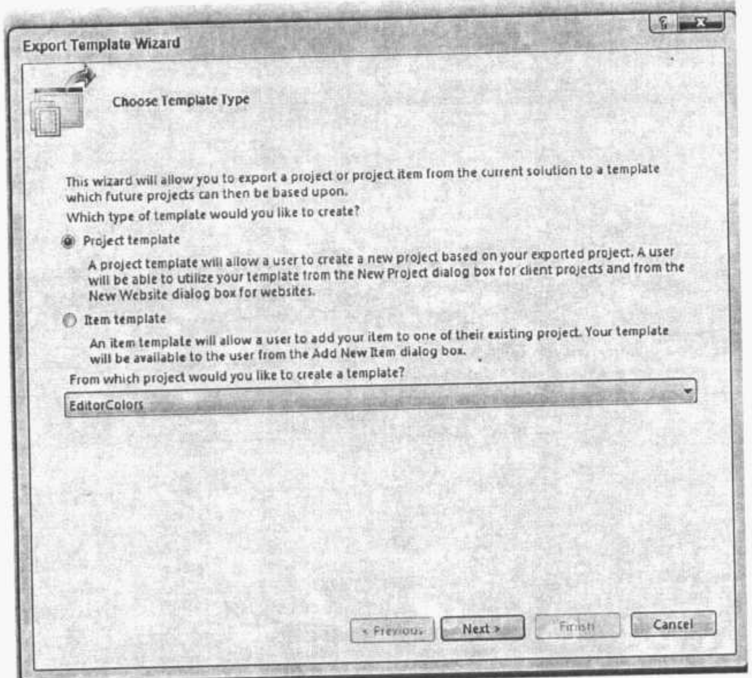
هذا الأمر يستخدم في حفظ الملف الحالي المفتوح في البرنامج باسم جديد و في مكان جديد .

أمر Save All

يستخدم هذا الأمر في حفظ التعديلات التي أجريت علي كل الملفات المفتوحة في البرنامج .

أمر Export Template

باختيار أمر Export Template الموجود في قائمة File يتم فتح المعالج الخاص بتصدير المشروع الحالي في شكل قالب جاهز كما يوضحه (شكل رقم ١-٦) , و يتيح لك هذا المربع الحواري حفظ المشروع الحالي أو أحد عناصر المشروع كقالب جاهز يمكنك استخدامه بعد ذلك في مشروعات أخري .



معالج تصدير القوالب الجاهزة
(شكل رقم ١-٦)

أمر Page Setup

يتيح لك أمر Page Setup و أمر Print ضبط أعدادات الطابعة و طباعة المستند الحالي المفتوح في البرنامج , فيمكنك علي سبيل المثال باستخدام هذه الأوامر طباعة الأكواد الأصلية التي قمت باستخدامها في عمل مشروع في حالة ما إذا قمت بإظهار هذه الأكواد , كما يمكنك أيضاً استخدام هذه الأوامر في طباعة الصور .

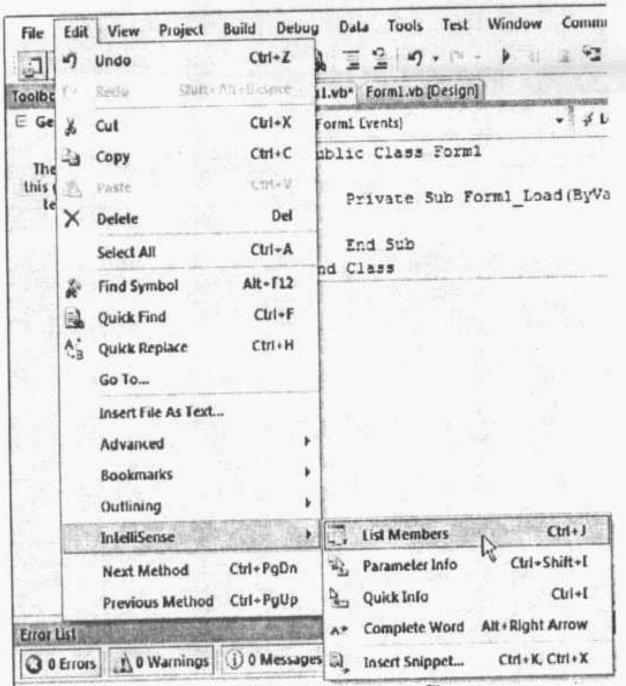
القائمة الفرعية Recent Files

القائمة الفرعية Recent Files و القائمة الفرعية Recent Projects تتيحان لك إعادة فتح أخر الملفات أو أخر المشروعات التي تم فتحها في البرنامج .

قائمة Edit

تحتوي قائمة Edit الموضحة في (شكل رقم ٢-٧) علي مجموعة الأوامر التي تستخدم في التعامل مع النصوص و العناصر الأخرى في المشروع أو الملف المفتوح في البرنامج .





قائمة Edit
(شكل رقم ١-٧)

و تحتوي هذه القائمة علي أوامر أساسية , مثل أمر التراجع Undo و التراجع عن التراجع Redo و القص Cut و النسخ Copy و اللصق Past و أخر الإلغاء Delete كما تحتوي هذه القائمة علي أوامر أخري كما يلي :

أمر Find Symbol

يتيح لك هذا الأمر البحث في المشروع عن مكونات البرامج كالمتغيرات و الثوابت .

أمر Quick Find

باختيار أمر Quick Find من قائمة Edit يظهر المربع الحواري الخاص بالبحث عن النصوص أو الكلمات في المشروع المفتوح في البرنامج , و من القائمة المنسدلة الخاصة بالبحث يمكنك تحديد ما إذا كان البحث يقتصر علي المستند الحالي النشط في البرنامج أم يشتمل علي كل المستندات المفتوحة في البرنامج , و من هذه القائمة أيضاً يمكنك اختيار البحث في مقطع محدد من هذا المستند , و من هذه القائمة أيضاً يمكنك اختيار أن يتم البحث عن كل الكلمات المفتاحية للبحث أو عن أي من هذه الكلمات فقط .

أمر Quick Replace

باختيار أمر Quick Replace من قائمة Edit يظهر نفس المربع الحواري الخاص بالبحث , و لكن في هذه الحالة يمكنك البحث و الاستبدال في النص من خلال ظهور حقل إضافي في هذا المربع الحواري خاص باستبدال الكلمات التي تبحث عنها بكلمات أخري .

أمر Go To

من خلال هذا الأمر يمكنك الانتقال إلي رقم سطر معين في المستند

الحالي .

أمر Insert File As Text

من خلال هذا الأمر يمكنك اختيار ملف نصي و أدراج ما به من نصوص في المكان الذي تريده في المستند .

القائمة الفرعية Advanced

تحتوي القائمة الفرعية Advanced الموجودة في قائمة Edit علي الأوامر التي تستخدم في تنسيق النصوص مثل استخدام الكلمات المرتفعة و الكلمات المنخفضة و التحكم في التفاف النص و كتابة التعليقات علي الأكواد .

القائمة الفرعية Bokmarks

تحتوي القائمة الفرعية Bokmarks الموجودة في قائمة Edit علي مجموعة الأوامر التي تستخدم في إضافة أو إزالة تأشيرات علي المستندات .

القائمة الفرعية Outlining

تحتوي هذه القائمة الفرعية علي الأوامر المتعلقة بتمديد أو تقليص جزء من الكود .

القائمة الفرعية Intellisense

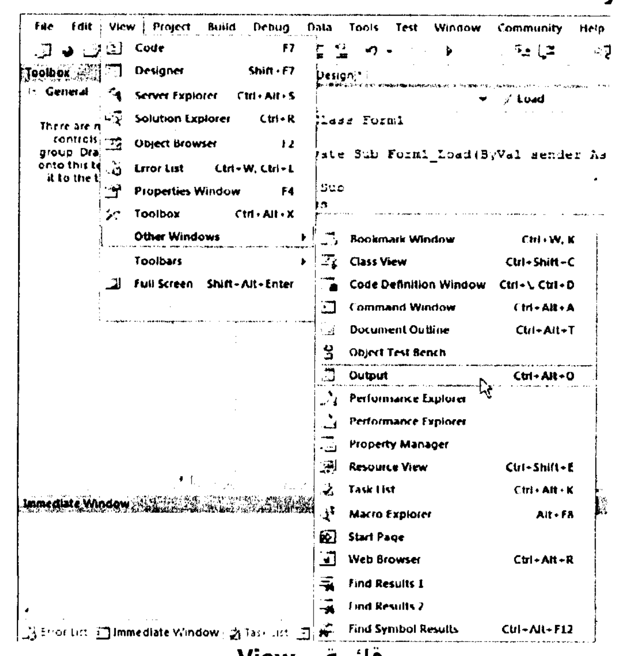
تستخدم الأوامر الموجودة في هذه القائمة في الوصول إلي مكونات الكائن و عرض هذه المكونات مثل الأحداث و الفئات .

أمر Next Method

يستخدم أمر Next Method و أمر Previous Method في التنقل بين مكونات الكائن البرمجي الموجود في المستند الحالي .

قائمة View

تحتوي قائمة View الموضحة في (شكل رقم ١-٨) علي مجموعة الأوامر الخاصة بإظهار و أخفاء عناصر واجهة البرنامج مثل النوافذ و الأشرطة الأدوات .



قائمة View

(شكل رقم ١-٨)



أمر Code

يستخدم أمر Code الموجود في قائمة View في فتح الملفات في نافذة محرر الأكواد , فعلي سبيل المثال إذا ما أردت فتح الملف الذي يحتوي علي الأكواد الخاصة بالنموذج , يمكنك تحديد هذا الملف في المتصفح الموجود في واجهة البرنامج و اختيار Code من قائمة View ليتم فتح هذا الملف في محرر الأكواد .

أمر Designer

يستخدم أمر Designer الموجود في قائمة View في فتح ملف الصور المحدد في متصفح البرنامج في محرر الصور , و في حالة ما إذا كان الملف المحدد غير صورة فإن هذا الأمر يختفي من القائمة .

الأوامر الخاصة بإظهار القوائم

مجموعة الأوامر الموجودة في قائمة View بعد أمر Designer مثل أمر Server Explorer و أمر Solution Explorer و أمر Object Browser و أمر Error List و أمر Properties Window و أمر Toolbox هذه الأوامر مخصصة لإظهار النوافذ الخاصة بكل أمر من هذه الأوامر علي حسب أسم الأمر .

القائمة الفرعية Other Window

الأوامر الموجودة في القائمة الفرعية Other Window الموجودة في قائمة View تستخدم في إظهار النوافذ الأخرى الغير مخصص لها أوامر في قائمة View نفسها

القائمة الفرعية Toolbars

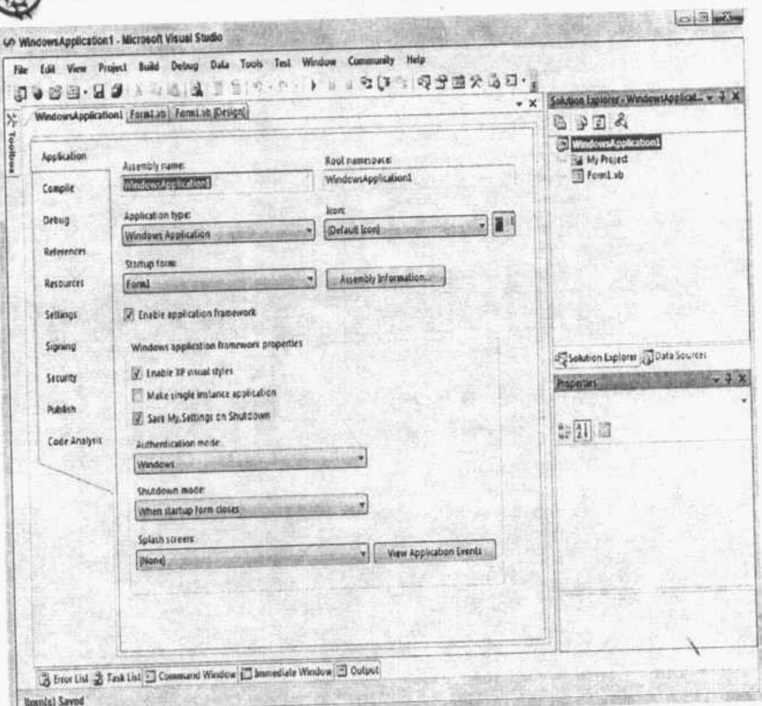
تستخدم الأوامر الموجودة في القائمة الفرعية Toolbars في إظهار و إخفاء أشرطة الأدوات , أو ضم هذه الأشرطة مع بعضها البعض .

أمر Full Screen

يستخدم أمر Full Screen في أخفاء كل أشرطة الأدوات و النوافذ فيما عدا نوافذ المحرر الذي تقوم باستخدامه , و يقوم هذا الأمر أيضاً بإخفاء شريط المهام بغرض توسيع مساحة العرض بقدر المستطاع للتمكن من التعامل مع الملفات المفتوحة في واجهة البرنامج , و باستخدام هذا الأمر يظهر زر صغير بالنقر عليه يمكنك الخروج من نمط عرض ملئ الشاشة .

أمر Property Pages

باختيار هذا الأمر يتم إظهار صفحات الخواص المتعلقة بالعناصر المحددة في متصفح البرنامج كما يوضحها (شكل رقم ١-٩) .



صفحات الخواص
(شكل رقم ١-٩)

قائمة Project

تحتوي قائمة Porject الموضحة في (شكل رقم ١-١٠) علي مجموعة الأوامر الخاصة بإضافة و إزالة عناصر إلي المشروع الحالي المفتوح في البرنامج , و تتوقف الأوامر التي تظهر في هذه القائمة علي العنصر المحدد حالياً .



Project | Build | Debug | Data | Tools | Test

- Add Windows Form...
- Add User Control...
- Add Component...
- Add Module...
- Add Class...
- Add New Item... Ctrl+Shift+A
- Add Existing Item... Ctrl+D
- Add New Distributed System Diagram...
- Exclude From Project
- Show All Files
- Add Reference...
- Add Web Reference...
- WindowsApplication1 Properties...

قائمة Project

(شكل رقم ١-١٠)

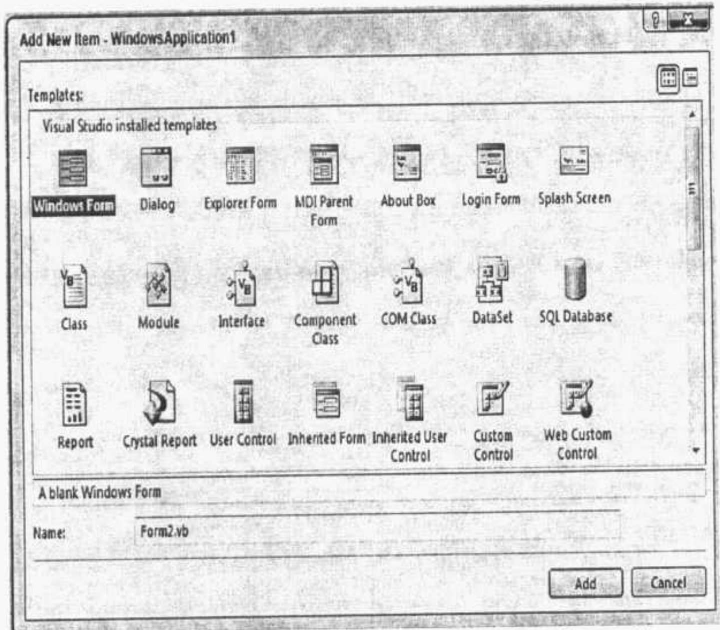
و فيما يلي توضيح للأوامر الموجودة في هذه القائمة .

مجموعة أوامر Add New

مجموعة الأوامر الأولي في قائمة Project و التي تبدأ بكلمة Add New تستخدم في إضافة عناصر إلي المشروع الحالي المفتوح في البرنامج , مثل أمر Add Window Form الذي يستخدم في إضافة نموذج , و أمر Add User Control الذي يستخدم لإضافة أدوات تحكم للمستخدم .

أمر Add New Item

باختيار أمر Add New Item من قائمة Project يظهر المربع الحواري الخاص بإضافة عناصر إلي المشروع كما يوضحه (شكل رقم ١-١١) , و الذي يتيح لك اختيار العناصر التي تريد أضافتها إلي المشروع مثل النصوص و الصور و الفئات .



المربع الحواري الخاص بإضافة عناصر إلي المشروع
(شكل رقم ١-١١)

أمر Add Existing Item

يتيح لك أمر Add Existing Item تصفح الملفات لاختيار العنصر الذي

تقوم بإضافته إلي المشروع .

أمر Exlude From Project

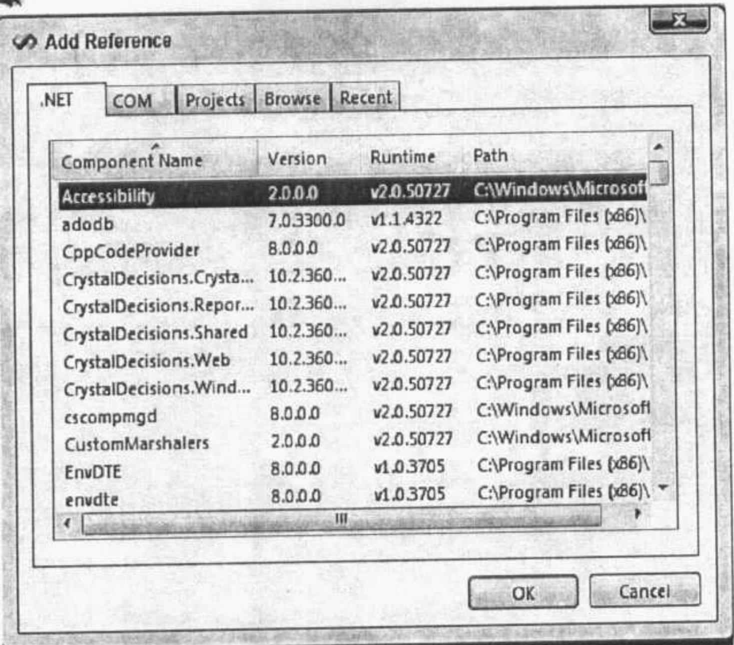
يقوم هذا الأمر بإزالة العناصر المحددة من المشروع , و هذا الأمر لا يقوم بإزالة الملف الخاص بالعنصر الذي قام بإزالته و أنما يقوم بإزالة العنصر من المشروع فقط .

أمر Show All File

يقوم هذا الأمر بإظهار كل الملفات في المتصفح الخاص بالبرنامج بما فيها الملفات المخفية و الملفات الأصلية المتعلقة بالنماذج و الفئات المخفية و الملفات الخاصة بالكائن البرمجي التي يتم إنشاؤها تلقائيا بواسطة البرنامج عندما يقوم بتشغيل المشروعات , و هذه الملفات مخفيه لأنك لن تحتاج إلي التعامل معها و لذلك قام فيجوال أستوديو بإخفائها , و يتم إظهار هذه الملفات باختيار الأمر من القائمة , و يتم إخفاؤها مرة أخري باختيار نفس الأمر مرة أخري .

أمر Add Reference

يقوم هذا الأمر بإظهار المربع الحواري الموضح في (شكل رقم ١-١٢) , و الذي يتيح لك اختيار عناصر برمجة خارجية عن المشروع و ربطها بالمشروع



المربع الحواري الخاص بربط عناصر خارجية بالمشروع

(شكل رقم ١-١٢)

و من خلال المربع الحواري الموضح في الشكل السابق يمكنك اختيار أكثر من عنصر خارجي عن المشروع لربطه بالمشروع و ذلك عن طريق تحديد أكثر من عنصر في المربع الحواري بالضغط علي مفتاح Shift أو مفتاح Ctrl في لوحة المفاتيح أثناء تحديد هذه العناصر في المربع الحواري .

أمر Add Web Reference

باختيار أمر Add Web Reference من قائمة Project يتم فتح المربع

الحواري الموضح في (شكل رقم ١-١٣) , و الخاص بالبحث عن خدمات الويب التي يمكن الربط إليها من المشروع الفتوح بالبرنامج بحيث يمكن استخدامها في موقع الويب الذي تقوم بإنشائه .

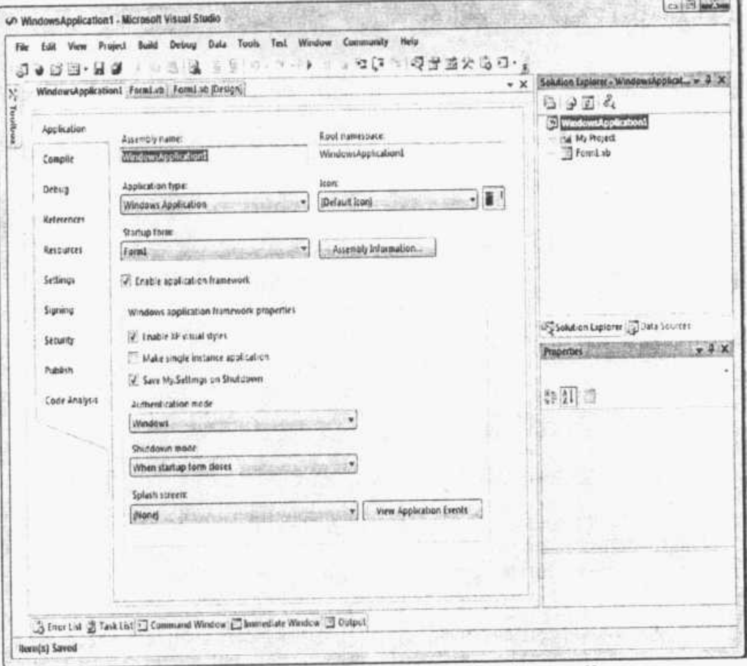
المربع الحواري الخاص بالربط إلي خدمات الويب

(شكل رقم ١-١٣)

أمر WindowsApplication 1 Properties

يقوم هذا الأمر بإظهار المربع الحواري الخاص بعرض صفحات الخواص كما هو موضح في (شكل رقم ١-١٤) .





المربع الحواري الخاص بتطبيقات الويندوز

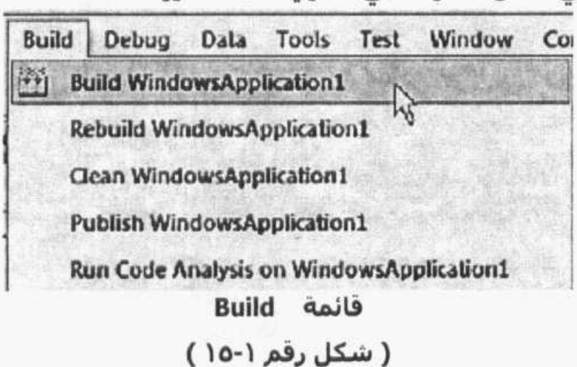
(شكل رقم ١-١٤)

و لا تظهر كل القوائم في المربع الحواري الموضح في الشكل السابق بشكل افتراضي و لكن هناك قوائم تم إخفاؤها لتوسيع مساحة العرض مثل قائمة Error List وقائمة Task List وقائمة Command Window وغيرها, و يمكن إظهار هذه القوائم المخفية بالنقر علي الأزرار الخاصة بأي من هذه القوائم في الشريط الموجود بأسفل واجهة البرنامج كما يوضحها الشكل السابق .

و في المربع الحواري الموضح في الشكل السابق تختلف القوائم التي تظهر في الجزء الأيمن من المربع الحواري علي حسب التطبيق الذي تقوم باختياره من قائمة التطبيقات الموجودة في يسار المربع الحواري .

قائمة Build

تحتوي قائمة Build كما يوضحها (شكل رقم ١-١٥) علي مجموعة الأوامر التي تستخدم في تشغيل الأكواد التي تحتوي عليها المشروعات .



قائمة Build

(شكل رقم ١-١٥)

و تحتوي هذه القائمة علي الأوامر التالية :

أمر Build WindowsApplication1

يقوم هذا الأمر بتفسير الأكواد و تشغيل المشروع الحالي عن طريق فحص ملفات المشروع للتعرف علي ما إذا كان هناك تعديلات تمت علي هذه الملفات منذ أخر مرة تم تشغيل المشروع فيها لتحديث هذه الملفات بهذه التعديلات .



أمر Rebuild WindowsAppliction 1

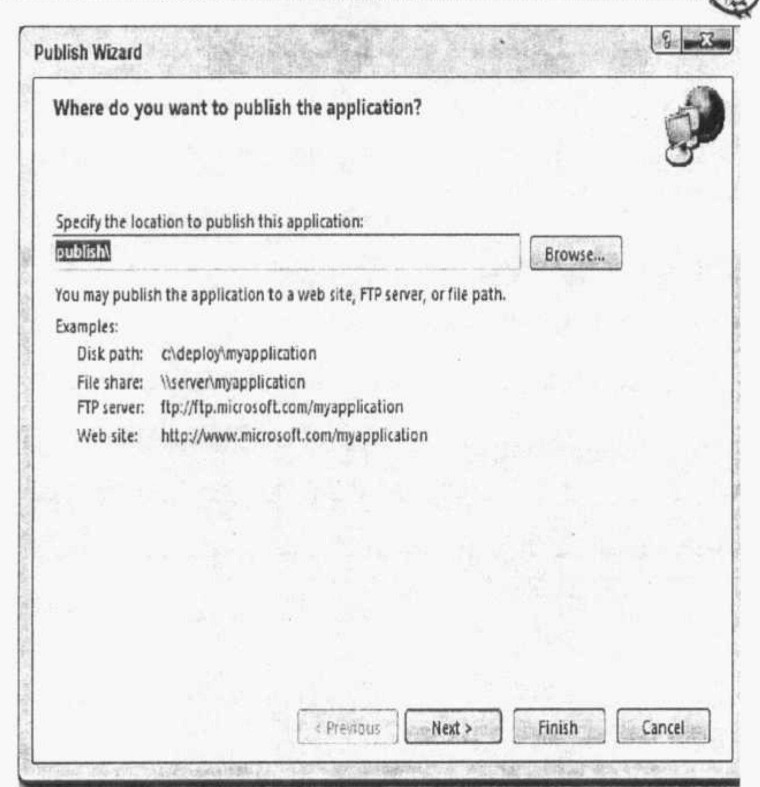
يقوم هذا الأمر بإعادة تفسير الأكواد و تشغيل المشروع الحالي من البداية عن طريق تفسير كل الأكواد الموجودة في كل ملفات المشروع و تنفيذها .

أمر Clean WindowsApplication 1

يستخدم هذا الأمر في إلغاء كل الملفات المؤقتة التي تم إنشاؤها أثناء تشغيل المشروع و يترك فقط الملفات الأصلية الخاصة بالمشروع و ملف المشروع النهائي بامتداد exe. و الملفات المرفقة بامتداد dll.

أمر Publish WindowAplication 1

يقوم هذا الأمر بفتح معالج إخراج المشروع بحيث يكون المشروع متاح لوضعه في ملفات أو نقله عبر شبكة الإنترنت أو عبر الشبكة المحلية , كما هو موضح في (شكل رقم ١-١٦) .



معالج إخراج المشروع
(شكل رقم ١-١٦)

أمر Run Conde Analysis on WindowsApplication1

يقوم هذا الأمر بتحليل المشروع الذي تم تنفيذه و كتابة تقرير عن الأخطاء أو المشاكل التي ربما تتواجد في المشروع كما يوضحه (شكل رقم ١-١٧)



Error List

0 Errors | 10 Warnings | 0 Messages

	Description	File	Line	Column	Project
1	CA2209 : Microsoft.Usage : No valid permission requests were found for assembly 'Factorial'. You should always specify the minimum security permissions using SecurityAction.RequestMinimum.				Factorial
2	CA2210 : Microsoft.Design : Sign 'Factorial' with a strong name key.				Factorial
3	CA1014 : Microsoft.Design : 'Factorial' should be marked with CLSCompliantAttribute and its value should be true.				Factorial
4	CA1709 : Microsoft.Naming : Correct the casing of type name 'frmFactorial'.	frmFactorial.Designer.>	2		Factorial
5	CA1305 : Microsoft.Globalization : frmFactorial.btnCalculate_Click (Object, EventArgs):Void makes a call to System.Double.ToString (System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Double.ToString (System.String, System.IFormatProvider).	frmFactorial.vb	7		Factorial
6	CA1305 : Microsoft.Globalization : frmFactorial.btnCalculate_Click (Object, EventArgs):Void makes a call to System.Int32.Parse (System.String) that does not explicitly provide an IFormatProvider. This should be replaced with a call to System.Int32.Parse (System.String, System.IFormatProvider).	frmFactorial.vb	4		Factorial
7	CA1303 : Microsoft.Globalization : frmFactorial.InitializeComponent ():Void passes a literal as parameter 1 of a call to Control.set_Text (String):Void. Retrieve the following string argument from a resource table instead: 'Calculate'	frmFactorial.Designer.>	35		Factorial
8	CA1303 : Microsoft.Globalization : frmFactorial.InitializeComponent ():Void passes a literal as parameter 1 of a call to Control.set_Text (String):Void. Retrieve the following string argument from a resource table instead: 'Factorial'	frmFactorial.Designer.>	78		Factorial
9	CA1303 : Microsoft.Globalization : frmFactorial.InitializeComponent ():Void passes a literal as parameter 1 of a call to Control.set_Text (String):Void. Retrieve the following string argument from a resource	frmFactorial.Designer.>	45		Factorial

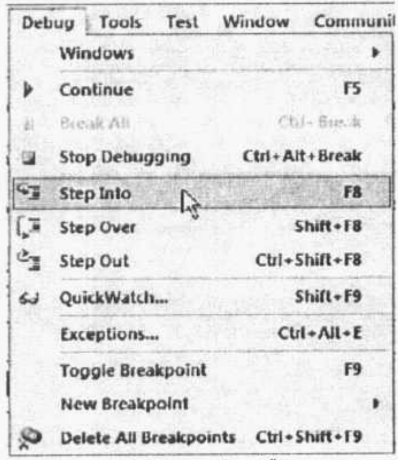
Error List | Task List | Command Window | Immediate Window | Output

التقرير عن الأخطاء الموجودة في المشروع
(شكل رقم ١-١٧)

قائمة Debug

تحتوي قائمة Debug الموضحة في (شكل رقم ١-١٨) علي مجموعة الأوامر التي تساعد في تصحيح الأخطاء في المشروع , هذه الأوامر تقوم بتشغيل مصحح الأكواد الذي يقوم بعمل مسح شامل للأكواد الموجودة في المشروع و يقوم بتحديد نقاط التوقف BreakPoints و أزالتها , و يقوم أيضاً بتتبع تنفيذ الأكواد للتأكد من صحة هذه الأكواد .

و ظهور الأوامر في هذه القائمة يتوقف علي عدة عوامل مثل نوع الملف المفتوح الذي تقوم بتنقيته من الأخطاء و علي المكان في الملف الذي يقف فيه مؤشر الماوس و علي ما إذا السطر الذي يقف عنده مؤشر الماوس يحتوي علي نقاط توقف أم لا .



قائمة Debug

(شكل رقم ١-١٨)

و فيما يلي عرض للأوامر التي تحتوي عليها قائمة Debug

أمر Continue

يستخدم أمر Continue الموجود في قائمة Debug في تخطي نقاط التوقف الموجودة في المشروع , فباستخدام هذا الأمر يتم تنفيذ البرنامج حتى تقوم أنت بإيقافه .

أمر Break All

يستخدم هذا الأمر في إيقاف كل البرامج التي تعمل أثناء فترة تصحيح الأخطاء , و يستخدم هذا الأمر في إيقاف كل البرامج التي تقوم بتشغيلها , و لذلك فإن هذا الأمر مفيد في حالة قيامك باختبار أكثر من برنامج تعمل معاً

أمر Stop Depugging

يقوم هذا الأمر بإيقاف عملية تشغيل البرامج بغرض اكتشاف الأخطاء الموجودة بها , و يتم إيقاف البرامج التي يتم اختبارها بمجرد اختيار هذا الأمر بدون ترك فرصة لتنفيذ الأكواد الخاصة بإزالة الملفات المؤقتة .

أمر Step Into

هذا الأمر يجعل مكتشف الأخطاء يقوم بتشغيل الكود الحالي الذي يقوم بفحصه , و لو كان هذا الكود يشمل تشغيل دالة معينه أو احد الملحقات الأخرى للمشروع كالدوال مثلا , هذا الأمر يجعل مكتشف الأخطاء ينتقل إلي هذه الملحقات .

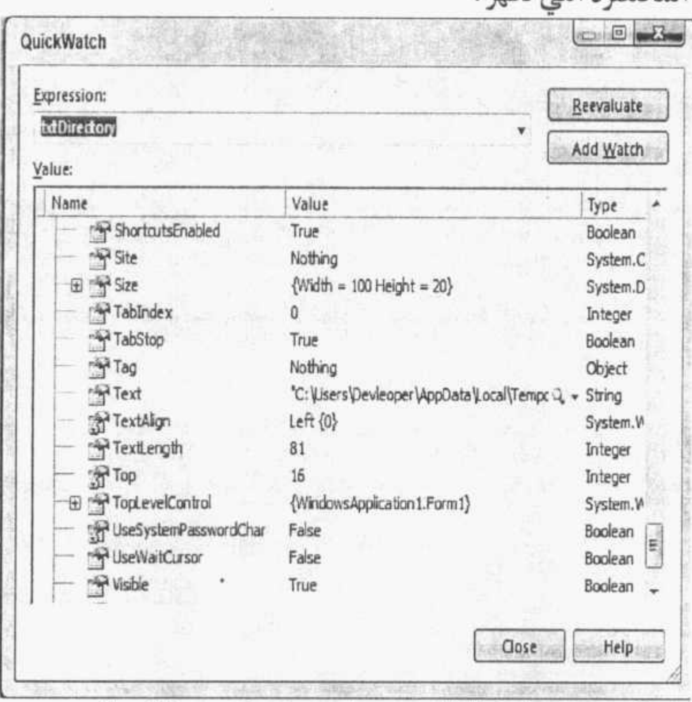
أمر Step Over

هذا الأمر يجعل عملية اكتشاف الأخطاء تقتصر علي الكود الحالي فقط و لا تمتد إلي ملحقات هذا الكود .

أمر QuickWatch

هذا الأمر يقوم بإظهار المربع الحواري الخاص بعرض معلومات عن كود الكائن المحدد كما هو موضح في (شكل رقم ١-١٩) , و من خلال هذا المربع الحواري يمكنك تعديل خواص الكائن البرمجي بالنقر بزر الماوس

الأيمن فوق الخاصية التي تريد تعديلها , و اختيار Edit Value من القائمة المختصرة التي تظهر .



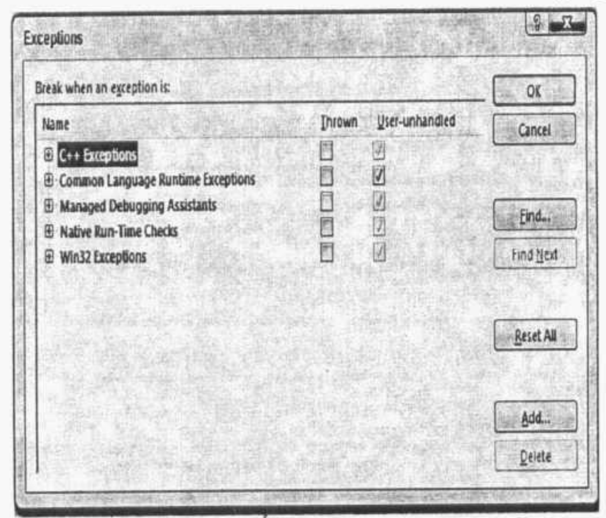
المربع الحواري الخاص بعرض معلومات عن الكود المحدد

(شكل رقم ١-١٩)

أمر Exceptions

يقوم هذا الأمر بعمل استثناء لاكتشاف بعض الأخطاء عن طريق إظهار المربع الحواري الموضح في (شكل رقم ١-٢٠) الذي يظهر بمجرد اختيار الأمر من القائمة , و في هذا المربع الحواري عندما تقوم بتحديد مربع

الخيار Thrown الموجود في المربع الحواري أمام أي نوع من أنواع الأخطاء الموجودة في قائمة أنواع الأخطاء الموجودة في المربع الحواري فإن مستكشف الأخطاء يقوم بتخطي هذا النوع من الأخطاء و يتوقف عن العمل عند اكتشاف هذا النوع من الخطأ , أما إذا ما قمت بتحديد مربع الاختيار User-unhandled الموجود أمام أي نوع من الأخطاء الموجودة في القائمة الموجودة في المربع الحواري فإن مستكشف الأخطاء يتوقف عندما يقابل خطأ من هذا النوع , و لكن في هذه الحالة فإن المستكشف لا يقوم بتشغيل الكود الخاص بالتعامل مع هذا النوع من الأخطاء .



المربع الحواري الخاص بأمر Exceptions

(شكل رقم ١-٢٠)

القائمة الفرعية New Breakpoint

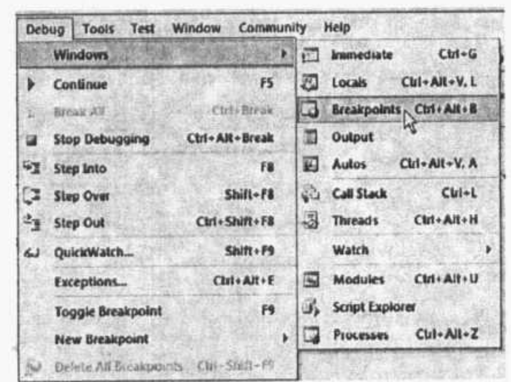
تحتوي هذه القائمة الفرعية الموجودة في قائمة Debug علي أمر التوقف عند الدوال , و هذا الأمر يقوم بفتح المربع الحواري الذي يتيح لك من خلاله تحديد الدوال التي يتوقف عندها البرنامج .

أمر Delete All Breakpoints

يقوم هذا الأمر بإزالة كل نقاط التوقف .

القائمة الفرعية Window

تحتوي هذه القائمة كما يوضحها (شكل رقم ١-٢١) علي الأوامر التي تقوم بعرض النوافذ الأخرى المستخدمة في تصحيح الأخطاء , و من الأوامر التي تحتوي عليها هذه القائمة أمر Immediate الذي يقوم بعرض نافذة Immediate و التي تمكنك من كتابة جمل برمجية بلغة فيجوال بيسيك .



القائمة الفرعية Windows

(شكل رقم ١-٢١)

و من الأوامر التي تحتوي عليها القائمة الفرعية Windows أيضاً الموجودة في قائمة Debug هو أمر Locals و الذي باختياره يتم عرض نافذة Locals الموضحة في (شكل رقم ١-٢٢) , و التي يعرض بها قيم المتغيرات المعرفة في المشروع , و يمكنك من خلال هذه النافذة تغيير قيم هذه المتغيرات بكتابة القيم الجديدة لهذه المتغيرات أو بالنقر فوق علامة + أو - الموجودة بيسار هذه القيم في القائمة كما يوضحها الشكل .

Locals

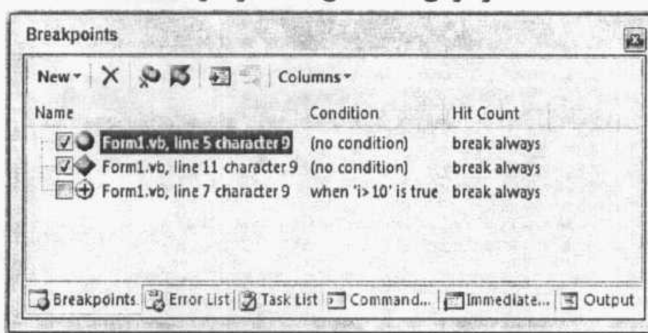
Name	Value	Type
⊞ Me	{WindowsApplication1.Form1}	WindowsApplication1.F
⊞ base_rate	{Length=12}	Single()
⊞ e	{System.EventArgs}	System.EventArgs
ex	Nothing	System.Exception
i	0	Integer
j	1337	Integer
⊞ sender	{WindowsApplication1.Form1}	Object
⊟ values	{Length=4}	Integer()
(0)	7635	Integer
(1)	2836	Integer
(2)	172	Integer
(3)	3871	Integer
x	1.32	Single
Y	2817.3	Single

قائمة Loacals

(شكل رقم ١-٢٢)

و من الأوامر الموجودة في القائمة الفرعيةWindows أيضاً أمر Breakpoints و الذي باختياره يتم عرض نافذة Breakpoints الموضحة في (شكل رقم ١-٢٣) , و التي يعرض بها كل نقاط التوقف و أماكن هذه النقاط و رقم

السطر الموجودة فيه , و بتحديد أو عدم تحديد مربع الاختيار الموجود بيسار أي من نقاط التوقف التي تظهر في النافذة يتم تشغيل أو تعطيل نقطة التوقف , كما يمكنك بالنقر بزر الماوس الأيمن فوق أي من نقاط التوقف الموجودة في النافذة يمكنك تعديل مكان هذه النقطة , كما يمكنك من خلال الأدوات الموجودة في شريط الأدوات الخاص بهذه النافذة إلغاء أي من هذه النقاط أو إنشاء نقاط جديدة أو تشغيل أو تعطيل هذه النقاط أو الذهاب للكود الموجود به نقطة التوقف , و من خلال القائمة المختصرة التي تظهر بالنقر بزر الماوس الأيمن فوق أي من نقاط التوقف يمكنك تغيير حالة هذه النقطة أو عرض حالتها من نشطة أو غير نشطة .



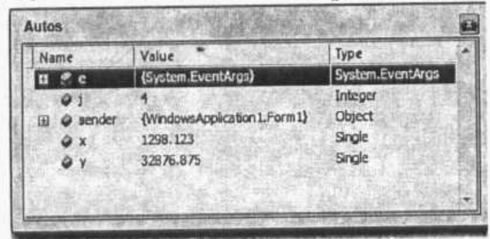
Name	Condition	Hit Count
Form1.vb, line 5 character 9	(no condition)	break always
Form1.vb, line 11 character 9	(no condition)	break always
Form1.vb, line 7 character 9	when 'i>10' is true	break always

نافذة **Breakpionts**

(شكل رقم ١-٣٣)

كما تحتوي قائمة Windows الفرعية من قائمة Debug أيضاً علي أمر Output و باختيار هذا الأمر من هذه القائمة الفرعية يقوم بعرض نافذة Output و التي يظهر بها تقرير عن مخرجات عملية تتبع الأخطاء , و تحتوي القائمة

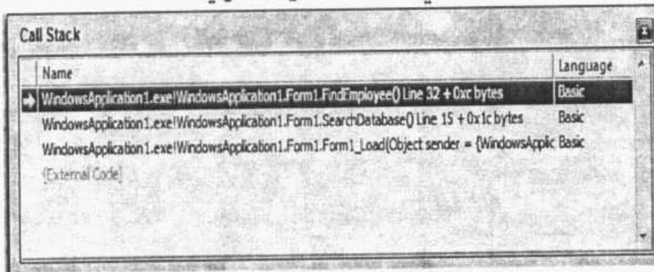
الفرعية Windows أيضاً علي أمر Autos و الذي باختياره من هذه القائمة يتم عرض نافذة Autos كما يوضحها (شكل رقم ١-٢٤) , و يظهر بهذه النافذة قيم المتغيرات الداخلية Local أو المتغيرات العامة Global الموجودة في السطر الحالي من البرنامج و في ثلاثة أسطر سابقة و ثلاثة أسطر تالية لهذا السطر



نافذة Autos

(شكل رقم ١-٢٤)

كما يظهر بالقائمة الفرعية Windows أيضا أمر Call Stack و هذا الأمر يقوم بعرض نافذة Call Sack الموضحة في (شكل رقم ١-٢٥) , و يظهر بهذه النافذة الأكواد المستخدمة في استدعاء أكواد أخري .



نافذة Call Stack

(شكل رقم ١-٢٥)

و تحتوي القائمة الفرعية Windows أيضاً علي أمر Threads الذي يعرض نافذة Threads الموضحة في (شكل رقم ١-٢٦) , و يعرض بهذه النافذة مسارات التنفيذ كل مسار مستقل عن الأخر و كل مسار من هذه المسارات يقوم بأداء عمل معين أي أن هذه المسارات جميعها التي تظهر في النافذة تقوم بتنفيذ مجموعة من الأعمال في أن واحد , و من خلال هذه النافذة يمكنك التحكم في أولوية تنفيذ هذه الأعمال , كما يمكنك تعطيل أي من هذه الأعمال عن طريق النقر بزر الماوس الأيمن فوق المسار الخاص به و اختيار Freeze من القائمة المختصرة التي تظهر , و يمكنك أعادة تشغيله باختيار Switch To من هذه القائمة .

Threads

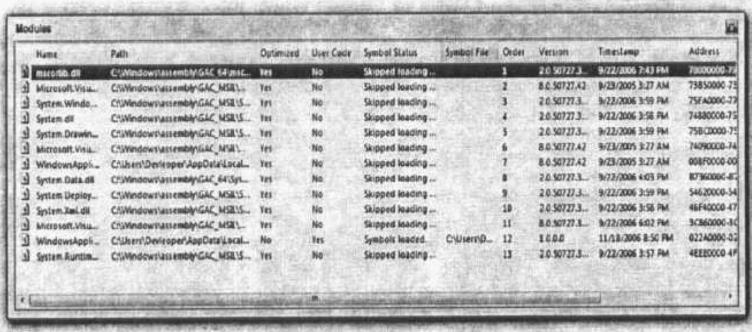
ID	Name	Location	Priority	Suspend
3028				0
3064	<No Name>		Highest	0
2684				0
2324	<No Name>		Normal	0
2776	<No Name>		Normal	0
2800	<No Name>		Normal	0
2272	.NET SystemEvents		Normal	0
704	<No Name>	WindowsApplication1.Form1.FindEmployee	Normal	0

نافذة Threads

(شكل رقم ١-٣٦)

و تحتوي قائمة Windows علي أمر Modules الذي يقوم بإظهار نافذة Modules الموضحة في (شكل رقم ١-٢٧) , و هذه النافذة تظهر معلومات عن ملفات DLL و ملفات EXE المستخدمة في البرنامج , و تحتوي هذه النافذة

علي أسماء الملفات المستخدمة في المشروع و المسار إلي هذه الملفات , و يمكن ترتيب هذه الملفات في النافذة علي حسب أي من معلومات الملفات التي تظهر في النافذة بالنقر بزر الماوس فوق العمود الخاص بهذه المعلومة , فعلي سبيل المثال يمكن ترتيب هذه الملفات علي حسب الاسم بالنقر علي عمود Name في النافذة .



نافذة Modules

(شكل رقم ١-٢٧)

كما تحتوي القائمة الفرعية Windows أيضاً علي أمر Script Explorer و الذي يقوم بعرض نافذة Script Explorer و التي تمكنك من تصحيح الأخطاء الموجودة في الأكواد المكتوبة باستخدام فيجوال بيسيك أو جافا سكريبت , كما تحتوي هذه القائمة أيضاً علي أمر Process الذي يقوم بعرض نافذة Process و التي تعرض الإجراءات التي قام بها البرنامج منذ أن قمت بفتحه في المرة الحالية المفتوح فيها البرنامج .

نافذة Breakpoints

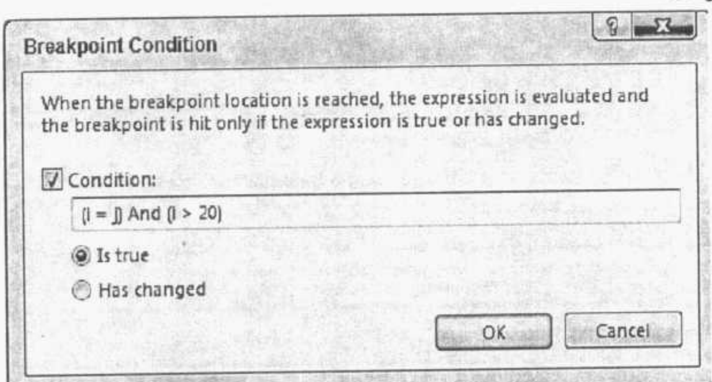
نقطة التوقف Breakpoint هي سطر برمجي قمت أنت بعمله يحتوي علي كود هذا الكود يقوم بإيقاف تنفيذ البرنامج , بحيث عندما يصل البرنامج إلي هذا الكود تتوقف عملية تنفيذ البرنامج , و يقوم فيجوال أستوديو بعرض الكود الخاص بنقطة التوقف في نافذة محرر الأكواد , و هذا العرض يتيح لك فحص الكود أو تحديد قيم للمتغيرات الموجودة في الكود .

و تعرض نافذة نقاط التوقف Breakpoints Window قائمة بكل نقاط التوقف التي قمت بعملها في البرنامج , و بذلك فهذه النافذة تسهل الوصول إلي نقاط التوقف الموجودة في البرنامج , و يحتوي فيجوال أستوديو علي أوامر تمكنك من تشغيل أو تعطيل نقاط التوقف أو إزالة هذه النقاط .

و تتضمن عملية تصحيح الأخطاء وضع علامات التعليق علي الأكواد التي تحتوي علي نقاط توقف لتعطيل هذه النقاط بحيث لا تعمل هذه النقاط , و بعد الانتهاء من عملية تصحيح الأخطاء فقد تحتاج إلي تشغيل بعض نقاط التوقف الموجودة في البرنامج و تسهل عليك نافذة نقاط التوقفBreakpoints Window هذه المهمة بعرضها كل نقاط التوقف الموجودة في البرنامج , و يمكنك عن طريق النقر المزدوج علي أي من نقاط التوقف في هذه النافذة الوصول إلي الكود الخاص بهذه النقطة بالطريقة التي تتيح لك تعديل هذا الكود , كما تتيح لك نافذة نقاط التوقف أيضاً تعديل هذه النقاط من خلال النافذة , فيمكنك من خلال النافذة تعطيل أو تشغيل هذه النقاط , و من خلال هذه النافذة أيضاً



يمكنك إزالة أي من نقاط التوقف الموجودة في البرنامج أو إزالتها كلها , فبالنقر بزر الماوس الأيمن فوق أي من نقاط التوقف في النافذة و اختيار Condition من القائمة المختصرة التي تظهر يظهر بذلك المربع الحواري الموضح في (شكل رقم ١ – ٢٨) , و من خلال هذا المربع الحواري يمكنك التحكم تنشيط أو تعطيل نقطة التوقف عن طريق تحديد شروط توقف البرنامج أو تخطيه لنقطة التوقف .

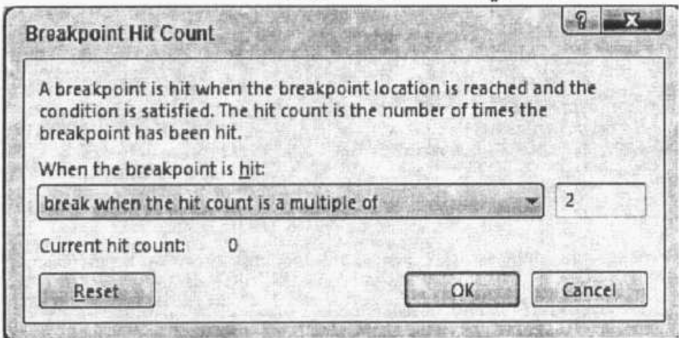


المربع الحواري الخاص بتنشيط أو تعطيل نقطة التوقف

(شكل رقم ١-٢٨)

و بالنقر بزر الماوس الأيمن فوق نقطة التوقف في النافذة و اختيار Hit Count من القائمة المختصرة التي تظهر يظهر بذلك المربع الحواري الموضح في (شكل رقم ١–٢٩) , و هذا المربع الحواري عبارة عن عداد لنقاط التوقف التي مر عليها البرنامج أثناء عملية تشغيله , فكل مرة يقوم بها البرنامج بالمرور علي نقطة توقف يتم زيادة الرقم الموجود في هذا المربع الحواري , و من

خلال هذا المربع الحواري يمكنك تشغيل أو تعطيل نقطة التوقف بناء علي قيمة العداد الموجود في هذا المربع الحواري .



المربع الحواري الخاص بعدد نقاط التوقف

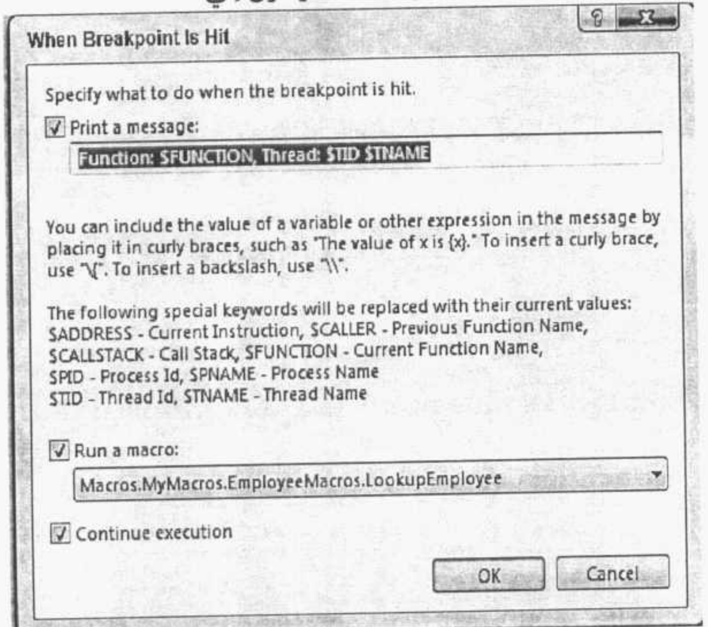
(شكل رقم ١-٢٩)

فعلي سبيل المثال من خلال هذا المربع الحواري يمكنك اختيار توقف البرنامج عن التنفيذ عندما يصل إلي نقطة التوقف رقم ٤ أو ٥ مثلاً .

و بالنقر بزر الماوس الأيمن فوق نقطة التوقف و اختيار When Hit من القائمة المختصرة التي تظهر , يتم عرض المربع الحواري الموضح في (شكل رقم ١-٣٠) , و من خلال هذا المربع الحواري يمكنك التحكم في ما يحدث عندما يصل البرنامج أثناء تنفيذه إلي نقطة التوقف , و بتنشيط مربع اختيار Print a message في المربع الحواري يقوم البرنامج بإظهار رسالة في نافذة Output Window عند وصولة إلي هذه النقطة , و بتنشيط مربع اختيار Run amacro في المربع الحواري يقوم البرنامج بتنفيذ أمر معين عند وصوله إلي هذه النقطة , و بتنشيط مربع اختيار Continue execution في المربع الحواري



يقوم البرنامج بالاستمرار بدون توقف عند الوصول إلي هذه النقطة .



المربع الحواري الخاص بتحديد سلوك البرنامج عند الوصول إلي نقطة التوقف

(شكل رقم ١-٣٠)

نافذة Immediate

نافذة Immediate و نافذة Command تمكنك من تنفيذ أمر معين عندما يصل البرنامج إلي نقطة معينه أثناء عملية تصحيح الأخطاء , و من أهم الأوامر التي يمكن للبرنامج تنفيذها عند الوصول إلي نقطة معينه هو أمر الطباعة

- Debug . Print

فباستخدام هذا الأمر مثلاً علي النحو التالي :

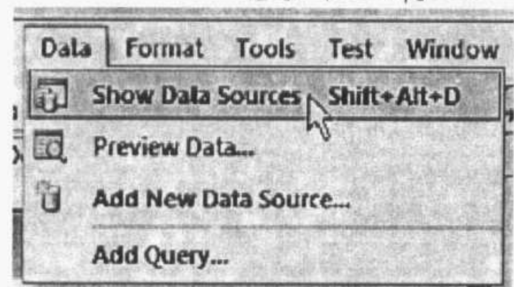
- Debug . print x

يقوم البرنامج بطباعة قيمة المتغير x في نافذة المخرجات Output Window كما يمكنك استخدام علامة الاستفهام ? بدلاً من أمر الطباعة Depug . Print كما يلي :

- >? x

قائمة Data

تحتوي قائمة Data الموضحة في (شكل رقم ١-٣١) علي الأوامر التي تستخدم في التعامل مع البيانات , و بعض هذه الأوامر لا تظهر و لا يمكن استخدامها إلا في حالة ما إذا كنت تقوم بتصميم نموذج .



قائمة Data
(شكل رقم ١-٣١)

و تحتوي قائمة Data علي الأوامر التالي :

امر Show Data Sources

باختيار هذا الأمر من قائمة Data يتم إظهار نافذة Data Sources و في



هذه النافذة يمكنك التعامل مع المصادر الأصلية للبيانات مثل الجداول و الحقول .

أمر Preview Data

يستخدم هذا الأمر في تحميل البيانات إلي مجمع البيانات DataSet و يقوم بإظهارها في وقت التصميم .

أمر Add New Data Source

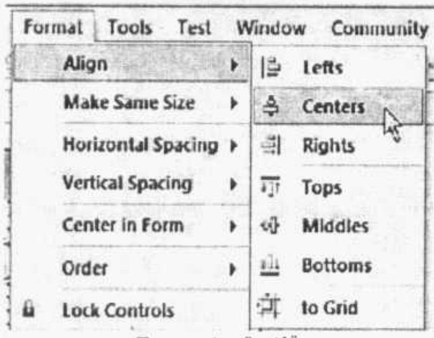
هذا الأمر يظهر معالج إضافة مصدر بيانات الذي يستخدم في إضافة البيانات إلي المشروع الذي تقوم بتنفيذه في البرنامج .

أمر Add Query

يستخدم هذا الأمر في فتح المربع الحواري الذي يستخدم في إضافة البيانات إلي النماذج .

قائمة Format

تحتوي قائمة Format الموضحة في (شكل رقم ١-٣٢) علي مجموعة الأوامر التي تستخدم في تنسيق النماذج .



قائمة Format
(شكل رقم ١-٣٢)

و فيما يلي الأوامر التي تحتوي عليها هذه القائمة .

القائمة الفرعية Align

تحتوي قائمة Align الفرعية من قائمة Format علي الأوامر التي تستخدم في تنسيق عناصر النموذج المحددة في النموذج , فتحتوي هذه القائمة علي أمر المحاذاة إلي اليسار Lefts و المحاذاة في الوسط Center و المحاذاة إلي اليمين Rights و المحاذاة لأعلي Tops و المحاذاة في المنتصف Middles و المحاذاة لأسفل Bottoms و المحاذاة للشبكة .

القائمة الفرعية Make Same Size

تحتوي قائمة Make Same Size الفرعية من قائمة Format علي مجموعة الأوامر التي تستخدم في تغيير حجم عناصر النموذج المحددة , مثل أمر Width و أمر Height لتحديد طول و عرض عنصر النموذج , و أمر Both و أمر Size to Gride الذي يقوم بتعديل حجم عنصر النموذج المحدد



بحيث يساوي مقدار حجم خلية الشبكة أو مضاعفاتها .

القائمة الفرعية Horizontal Sapcing

تحتوي هذه القائمة الفرعية من قائمة Format علي الأوامر التي تستخدم في تغيير المسافة الأفقية بين عناصر النموذج المحددة , مثل أمر Make Equal الذي يجعل المسافة الأفقية بين هذه العناصر متساوية , و أمر Increase و أمر Decrease اللذان يستخدمان في زيادة و نقص هذه المسافة .

القائمة الفرعية Verical Spacing

تحتوي هذه القائمة الفرعية من قائمة Format علي نفس الأوامر الموجودة في قائمة Horizontal Spacing و لكن الخاصة بالتحكم في المسافة الرأسية بين عناصر النموذج المحددة .

القائمة الفرعية Center in Form

تحتوي هذه القائمة الفرعية من قائمة Format علي مجموعة الأوامر التي تستخدم في توسيط عناصر النموذج مثل أمر Horizontally و أمر Vertically للتوسيط الأفقي و الرأسي لهذه العناصر .

القائمة الفرعية Order

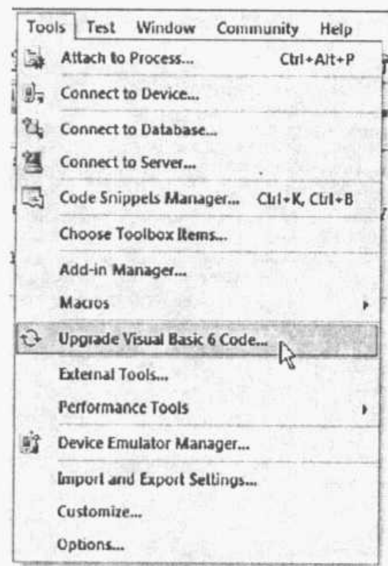
تحتوي القائمة الفرعية Order علي مجموعة الأوامر التي تستخدم في إرسال عناصر النموذج إلي الأمام و إلي الخلف أو إلي أعلي أو إلي أسفل .

أمر Lock Controls

يستخدم أمر Lock Controls في منع كل عناصر التحكم في النموذج من الحركة أو تغيير الحجم عن طريق السحب بزر الماوس .

قائمة Tools

تحتوي قائمة Tools الموضحة في (شكل رقم ١-٣٣) علي مجموعة من الأدوات .



قائمة Tools

(شكل رقم ١-٣٣)

و فيما يلي عرض للأوامر الموجودة في هذه القائمة .

أمر Attach to Process

يقوم هذا الأمر بإظهار المربع الحواري الذي يمكنك من استخدام مستكشف الأخطاء في العملية التي تقوم بها حالياً أو العملية المحددة وقت فتح المربع الحواري .

أمر Connect to Device

يستخدم هذا الأمر في الاتصال بأي أجهزة أو وسائط أخري كالكمبيوتر المحمول مثلاً التي قد تستخدمها في تجربة البرنامج .

أمر Connect to Database

هذا الأمر يقوم بعرض المربع الحواري الخاص بخواص الاتصال بقواعد البيانات , و من هذا المربع الحواري تستطيع تحديد قواعد البيانات التي تقوم بالأنصال بها .

أمر Connect to Server

يقوم هذا الأمر بعرض المربع الحواري الخاص بالأنصال ببرنامج قواعد البيانات .

أمر Code Snippets Manger

يستخدم هذا الأمر في إظهار المربع الحواري الخاص بإظهار الملفات الصغيرة التي تحتوي علي الأكواد , و الذي من خلاله يمكنك إضافة أو إزالة أي من هذه الملفات .

أمر Choose Toolbox Item

يقوم هذا الأمر بإظهار المربع الحواري الذي من خلاله تستطيع اختيار أي

من الأدوات التي تظهر في شريط الأدوات .

أمر Add-in Manger

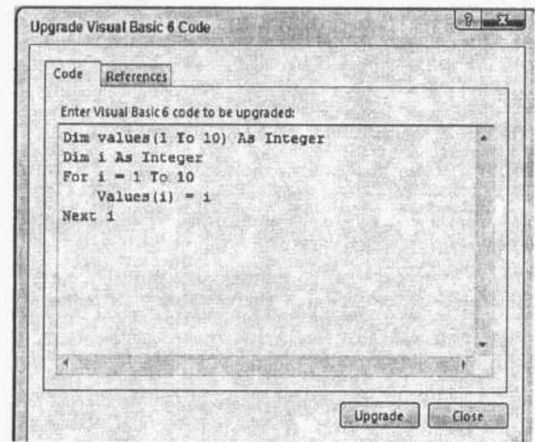
يقوم هذا الأمر بإظهار المربع الحواري الذي يستخدم في إدارة المشروعات الإضافية الموجودة علي الجهاز .

القائمة الفرعية Macros

تحتوي هذه القائمة الفرعية علي الأوامر التي تستخدم في إنشاء و تعديل و تنفيذ وحدات الماكرو .

أمر Upgrade Visual Basic 6 Code

يقوم هذا الأمر بعرض المربع الحواري الموضح في (شكل رقم ١-٣٤) , و الذي يستخدم في تحديث الأكواد المكتوبة بإصدار أقدم من 2005 إلي فيجوال بيسيك 2005



المربع الحواري الخاص بتحديث الأكواد
(شكل رقم ١-٣٤)



و لتحديث الأكواد باستخدام هذا المربع الحواري قم بكتابة الكود في المربع الحواري و اضغط زر Update بأسفل المربع الحواري .

أمر External Tools

يقوم هذا الأمر بإظهار المربع الحواري المستخدم في إضافة أو إزالة أوامر من قائمة Tools

أمر Device Emulation Manger

يقوم هذا الأمر بإظهار المربع الحواري الذي من خلاله تستطيع أن تتحكم في الوسائط الأخرى الموصلة بالجهاز , من حيث الاتصال أو عدم الاتصال بهذه الوسائط أو إغلاق هذه الوسائط .

أمر Import/Export Settings

يقوم هذا الأمر بإظهار المربع الحواري الخاص بحفظ الأعدادات الحالية لبرنامج Visual Studio أو العودة بهذه الأعدادات للوضع الافتراضي للبرنامج , أو الاحتفاظ بهذه الأعدادات أو استيراد أعدادات سابقة .

أمر Customize

يستخدم هذا الأمر في تعديل عناصر برنامج فيجوال أستوديو .

أمر Options

يستخدم هذا الأمر في تحديد خيارات واجهة البرنامج فيجوال أستوديو .

قائمة Test

تحتوي قائمة Test الموضحة في (شكل رقم ١-٣٥) علي مجموعة الأوامر الخاصة بالتحكم في أدوات الاختبار في البرنامج , و التي من خلالها يمكن عمل اختبار محدود أو اختبار شامل للمشروع الذي تقوم بتنفيذه , كما يمكنك من خلال هذه الأوامر أيضاً عمل اختبار يشابه إلي حد كبير الطريقة التي يقوم بها المستخدم النهائي للمشروع عند استخدامه المشروع الذي تقوم بتنفيذه .

Test | Window | Community | Help

- New Test...
- Load Metadata File...
- Create New Test List...
- Start Selected Test Project with Debugger — Shift+Alt+X
- Start Selected Test Project without Debugger — Ctrl+Shift+X
- Administer Test Controllers...
- Select Active Test Run Configuration ▸
- Edit Test Run Configurations ▸
- Windows ▸

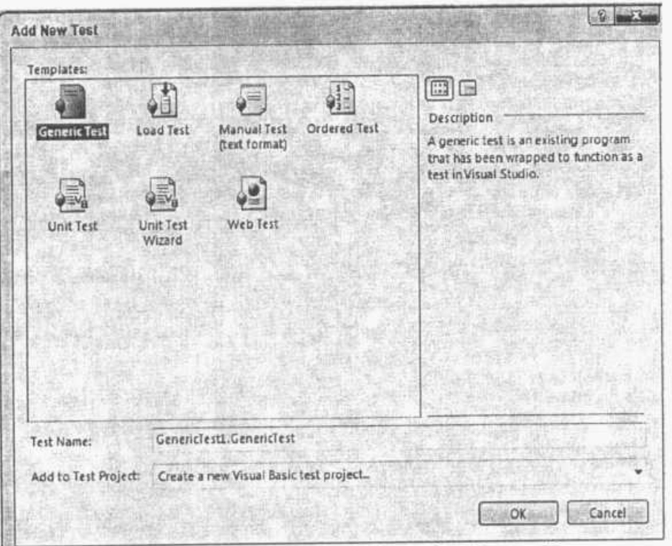
قائمة Test

(شكل رقم ١-٣٥)

و فيما يلي عرض لمجموعة الأوامر التي تحتوي عليها هذه القائمة .

أمر New Test

باختيار هذا الأمر من قائمة Tesst يظهر المربع الحواري الموضح في (شكل رقم ١-٣٦) , و الذي يتيح لك اختيار نوع الاختبار الذي تريد أن تقوم به .



المربع الحواري الخاص بتحديد نوع الاختبار
(شكل رقم ١-٣٦)

أمر Load Metadata File

يتيح لك هذا الأمر فتح ملفات الميتا داتا و التي تحتوي علي بيانات XML التي تقوم بوصف قوائم الاختبار .

أمر Create New Test List

من خلال هذا الأمر يمكنك عمل قائمة اختبار جديدة تضم مجموعة عمليات الاختبار المرتبطة ببعضها البعض و التي يتم تنفيذها معاً .

أمر Start Selected Test Project With Debugger

هذا الأمر يقوم ببدء تنفيذ الاختبار المحدد حالياً مع مستكشف الأخطاء .

أمر Start Selected Test Project Without Debugger

هذا الأمر يقوم ببدء تنفيذ الاختبار المحدد حاليا بدون مستكشف الأخطاء .

أمر Administer Test Controllers

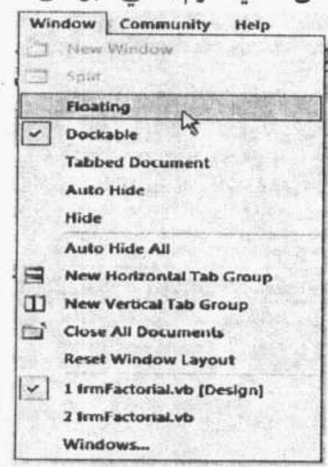
يتيح لك هذا الأمر إدارة أدوات التحكم في الاختبار .

أمر Select Active Test Run Configuration

هذا الأمر يتيح لك اختيار مؤشرات الاختبار الذي يجري حالياً مثل أسم الملف الذي يتم اختباره و عرض معلومات عن الاختبار .

قائمة Windows

تحتوي قائمة Window الموضحة في (شكل رقم ١ -٣٧) علي مجموعة الأوامر التي تتيح لك التحكم في نوافذ البرنامج , و تتوقف الأوامر التي تكون متاحة في النافذة علي نوع العمل الذي تقوم به في البرنامج .



قائمة Window
(شكل رقم ١-٣٧)



و فيما يلي عرضاً للأوامر التي تحتوي عليها قائمة Window

أمر New Window

يقوم هذا الأمر بإنشاء نافذة جديدة لعرض محتويات نافذة الكود الحالي .

أمر Split

يقوم هذا الأمر بتقسيم النافذة التي تحتوي علي الكود إلي قسمين لعرض الأجزاء المختلفة من الكود في وقت واحد , و باختيار هذا الأمر من القائمة يتحول إلي أمر Remove Split الذي يمكنك من إلغاء عملية تقسيم النافذة و إعادة النافذة لوضعها الأصلي مرة أخري .

أوامر Floating و Dockable و Tabbed Document

تقوم هذه الأوامر بالتحكم في الطريقة التي تظهر بها النوافذ الفرعية مثل نافذة Toolbox و نافذة Solution Explorer

أمر Auto Hide

باختيار أمر Auto Hide من قائمة Window يتم وضع النافذة في وضع الإخفاء التلقائي , و في هذا الوضع تختفي النافذة و لا يظهر منها سوي أسمها فقط , و بالنقر علي أسم النافذة تعود النافذة للظهور مرة أخري .

أمر Hide

يؤدي هذا الأمر إلي أخفاء النافذة تماماً و لا يظهر منها أي شيء .

أمر Auto Hide All

يقوم هذا الأمر بوضع كل النوافذ المساعدة في وضع الإخفاء التلقائي .

أمر New Horizontal Tab Group

يقوم هذا الأمر بتقسيم النافذة أفقياً بحيث تستطيع فتح مستندين متجاورين في النافذة في وقت واحد .

أمر New Vertical Tab Group

يقوم هذا الأمر بتقسيم النافذة رأسياً بحيث يمكنك فتح ملفين في النافذة في نفس الوقت أحدهما فوق الأخر .

أمر Close All Documents

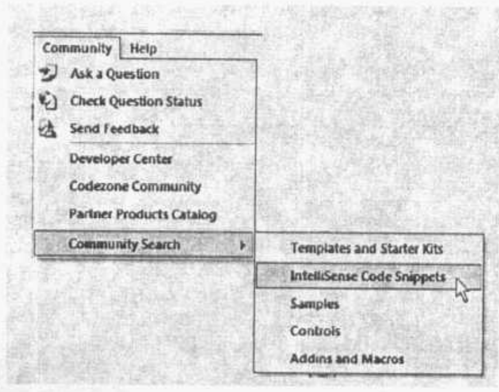
يؤدي هذا الأمر إلي إغلاق كل المستندات .

أمر Reset Window Layout

يقوم هذا الأمر بالعودة بالنوافذ إلي الوضع الافتراضي لها .

قائمة Community

تحتوي قائمة Community الموضحة في (شكل رقم ١-٣٨) علي مجموعة من الروابط لموقع شركة مايكروسوفت علي شبكة الإنترنت , و هذه الروابط لا تعمل إلا إذا كنت متصلاً بالشبكة .



قائمة Community
(شكل رقم ١-٣٨)





المكونات الأخرى لبرنامج Visual Studio

أنماط العرض
أشرطة الأدوات Toolbars

أنماط العرض

نمط عرض التصميم

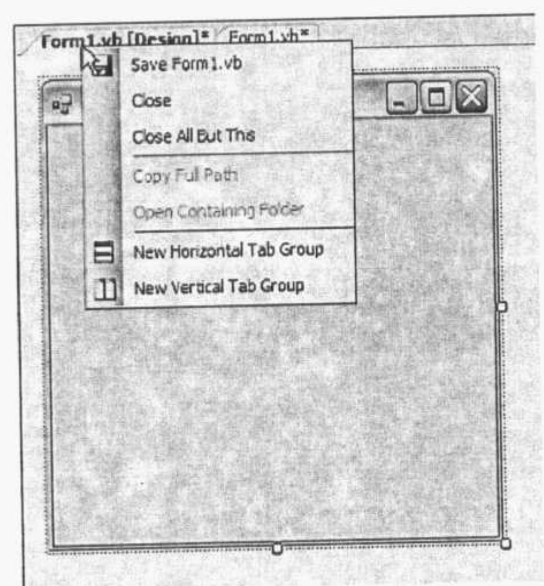
نمط عرض التصميم Designe View هو النمط الافتراضي الذي يعرضه فيجوال أستوديو عندما تقوم بفتح البرنامج في أول مرة , و يحتوي هذا النمط علي بيئة رسومية تحتوي علي أيقونات و أزرار و أدوات مرئية تستخدم في تنفيذ المشروع و ليس علي أكواد كما يعرض نمط عرض الأكواد , و يقوم فيجوال أستوديو بترجمة كل ما تقوم باختياره من أوامر موجودة في القوائم أو النقر علي أيقونات أو تحديد خيارات أو النقر علي أزرار يقوم بترجمة كل ذلك إلي أكواد .

و من خلال الأدوات الموجودة في نمط عرض التصميم يمكنك أداء الكثير من الأعمال في المشروع الذي تقوم ببنائه في البرنامج , فيمكنك إنشاء النماذج و أدوات التحكم الخاصة بهذه النماذج و تحديد خواصها , و يمكنك أيضاً من خلال هذه الأدوات التعامل مع قواعد البيانات و ملفات XML كما يمكنك أنشاء أجزاء المشروع الذي تقوم ببنائه من برامج و خلافه و ذلك دون أن تضطر إلي كتابة سطر برمجي واحد .

و يعتبر عرض التصميم هو العمل الأساسي لبرنامج فيجوال أستوديو حيث صمم البرنامج أساساً لقيام المستخدم بتنفيذ كل الأعمال التي يريدها من خلال الأدوات التي تحتوي عليها واجهة هذا النمط لعرض البرنامج .

الوصول إلي نمط عرض التصميم

في أعلي يسار واجهة البرنامج يوجد زر يسمي Design كما يوضح (شكل رقم ٣-١) , و تنشيط هذا الزر معناه أن البرنامج في نمط عرض التصميم و أن لم يكن كذلك فإنه بالنقر علي هذا الزر يتم وضع البرنامج في هذا النمط للعرض , و يمثل الشريط الرمادي الملفات المفتوحة حالياً في البرنامج و لكنها غير نشطه , و تشير علامة * الموجودة بجوار أسم الملف أن هذا الملف قمت بإجراء تعديلات عليه و لكن هذه التعديلات لم يتم حفظها .



الزر الخاص بنمط عرض التصميم
(شكل رقم ٣-١)

نمط عرض الأكواد

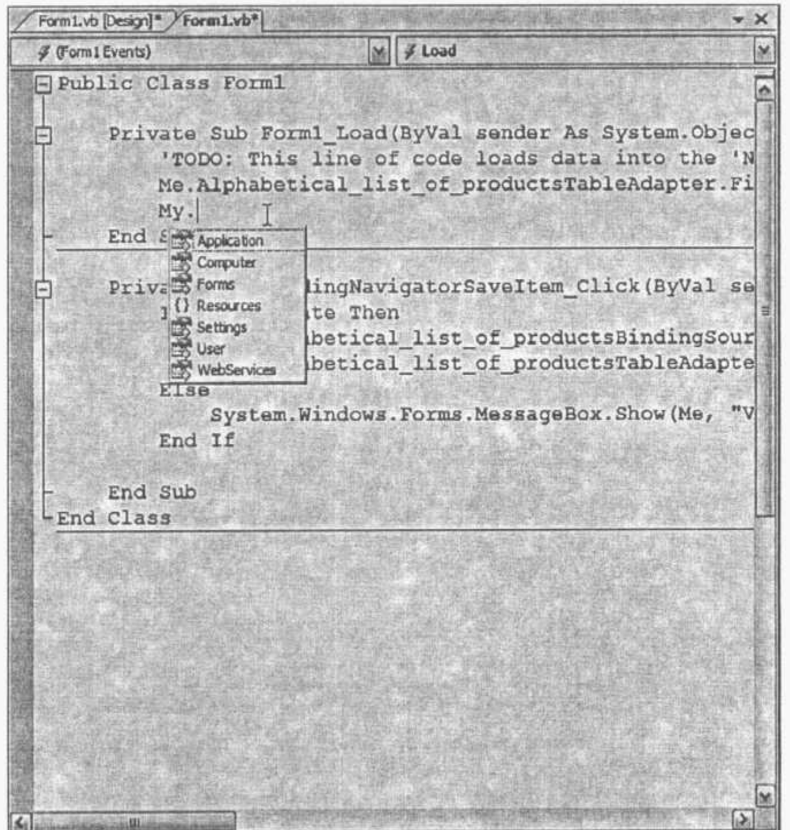
يحتوي برنامج فيجوال أستوديو إلي جانب نمط عرض التصميم علي العديد من الأدوات التي تمكنك من كتابة الأكواد بيدك أو تعديل الأكواد التي قام البرنامج بكتابتها نيابة عنك و التي تظهر باختيار نمط عرض الأكواد Code View و باختيار هذا النمط للعرض يختفي نمط عرض التصميم و يظهر هذا النمط للعرض في واجهة البرنامج .

و هناك أكثر من طريقة من خلالها يمكنك الوصول إلي نمط عرض الأكواد Code View فيمكنك الوصول إلي هذا النمط للعرض عن طريق النقر بزر الماوس الأمين فوق النموذج في نافذة Solution Explorer أو في نمط عرض التصميم و اختيار View Code من القائمة المختصرة التي تظهر , كما يمكنك أيضاً الوصول إلي نمط عرض الأكواد عن طريق النقر علي زر View Code الخاص بإظهار الأكواد الموجود في Solution Explorer كما يمكنك أيضاً الوصول إلي نمط عرض الأكواد بالنقر المزدوج علي أي عنصر من عناصر التحكم الموجودة في النموذج .

أداة مساعدة كتابة الأكواد

يحتوي نمط عرض الأكواد علي أداة Inellisense التي تساعدك عند قيامك بكتابة الأكواد كما يوضحها (شكل رقم ٣-٢) .





أداة المساعدة في كتابة الأكواد

(شكل رقم ٣-٢)

و لاستخدام أداة المساعدة في كتابة الأكواد كالموضحة في الشكل السابق , قم بفتح نمط عرض الأكواد بأحد الطرق السابق عرضها , و قم بكتابة أي عبارة في هذه الأكواد , قم كتابة My مثلاً لتقوم هذه الأداة بإظهار قائمة

مختصرة كالموضحة في الشكل السابق , تحتوي هذه القائمة علي كل الأكواد المقترحة التي يمكنك استخدامها بناء علي الكلمة التي قمت بكتابتها .

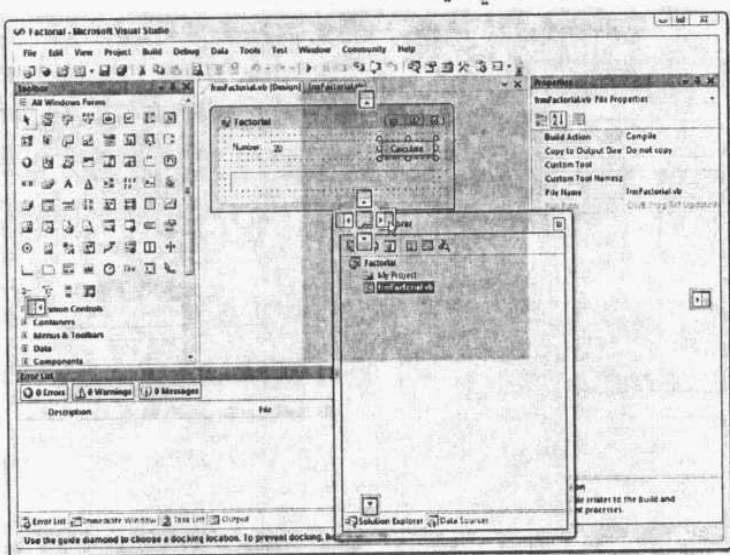
أشرطة الأدوات Toolbars

من السهل ترتيب أشرطة الأدوات التي يحتوي عليها برنامج Visual Studio و ذلك عن طريق سحب مقابض التحكم الرمادية الموجودة حول هذه الأشرطة و سحبها إلي المكان الذي تريد وضعها فيه , و إذا ما قمت بسحب أحد أشرطة الأدوات إلي حافة واجهة البرنامج , فأن هذه الأشرطة فإن هذه الأشرطة يتم تنكمش أما أفقياً في أعلي أو أسفل واجهة البرنامج , أو رأسياً في جانبي واجهة البرنامج , أما لو قمت بسحب أي من هذه الأشرطة بعيداً عن حافة واجهة البرنامج ففي هذه الحالة يكون الشريط حر الحركة بحيث يمكنك نقله في أي مكان في واجهة البرنامج , كما يمكنك أيضاً استخدام الأوامر الموجودة في القوائم الخاصة بإظهار و أخفاء أشرطة الأدوات ,

النوافذ الثانوية

يمكنك أيضاً إعادة تنسيق النوافذ الثانوية secondary Winows تماماً مثل أشرطة الأدوات , و ذلك عن طري سحب هذه النوافذ بزر الماوس من شريط العنوان الخاص بكل نافذة , و مع سحب النافذة بزر الماوس يظهر البرنامج أيقونة زرقاء صغيرة كالموضحة في (شكل رقم ٣-١) لتساعدك هذه الأيقونة في دمج النافذة مع مجموعة نوافذ أخري , و يظهر أيضاً أربعة

أيقونات قرب حواف واجهة البرنامج , و بإلقاء النافذة التي تقوم بسحبها عند أي من هذه الأيقونات الأربعة الموجودة في حواف واجهة البرنامج يتم محاذاة هذه النافذة التي تقوم بسحبها عند هذه الحافة , أما لو قمت بوضع النافذة التي تقوم بسحبها في أي مكان أخر بخلاف الأيقونات الأربعة التي تظهر عند حواف واجهة البرنامج ففي هذه الحالة تكون النافذة حرة الحركة يمكن وضعها في أي مكان .



ترتيب النوافذ الثانوية

(شكل رقم ٣-١)

أما إذا قمت بوضع النافذة في منطقة أساسية في واجهة البرنامج ففي هذه الحالة تصبح النافذة ثابتة في مكانها ولا يمكنك تحريكها من هذا المكان

إلا باستخدام أمر Floating من القوائم .

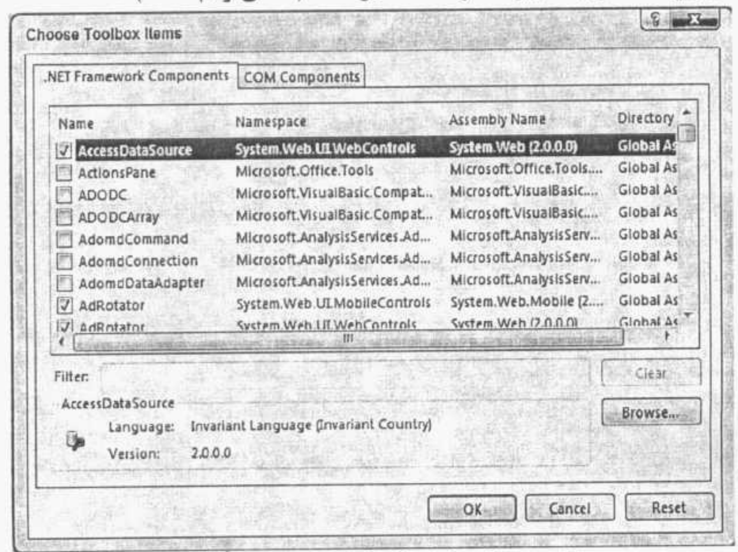
صندوق الأدوات

يتكون صندوق الأدوات Toolbox كما يوضحه (شكل رقم ٣-٢) من عدة أجزاء كل جزء من هذه الأجزاء يحتوي علي مجموعة من الأدوات تناسب التعامل مع المستند النشط في البرنامج , و الأدوات الموجودة في الصندوق كما يوضح الشكل مجمعة علي شكل أشرطة أدوات , و يمكنك التحكم في حجم صندوق الأدوات كبيراً أو صغيراً بحيث يظهر القدر اللازم من الأدوات التي تقوم باستخدامها في الوقت الحالية .

صندوق الأدوات
(شكل رقم ٣-٢)

و يمكنك ترتيب الأدوات في صندوق الأدوات بما يناسب احتياجاتك من خلال الأوامر الموجودة في القائمة المختصرة التي تظهر بالنقر بزر الماوس الأيمن علي أي من أشرطة الأدوات الموجودة في هذا الصندوق .

فباختيار أمر List View من هذه القائمة المختصرة يمكنك عرض الأدوات التي يحتوي عليها الشريط في شكل قائمة , و باختيار أمر Show All من هذه القائمة المختصرة يتم إظهار معظم و ليس كل الأدوات الموجودة في هذا الشريط , و باختيار أمر Choose Items من هذه القائمة يظهر المربع الحواري الخاص باختيار الأدوات كما يوضحه (شكل رقم ٣-٣) .



المربع الحواري الخاص باختيار الأدوات
(شكل رقم ٣-٣)

كما يمكنك ترتيب الأدوات في الشريط ترتيباً أبجدياً باختيار Sort Items



Alphabetically من القائمة المختصرة التي تظهر بالنقر بزر الماوس الأيمن فوق شريط الأدوات في صندوق الأدوات , و باختيار أمر Reset Toolbox من هذه القائمة يمكنك للعودة بالأدوات الموجودة في شريط الأدوات إلي الوضع الافتراضي لها , و باختيار أمر Add Tab يمكنك أنشاء شريط الأدوات جديد بداخل صندوق الأدوات و يمكنك استخدام هذا الشريط في وضع الأدوات المفضلة لديك و التي تستخدمها بشكل مستمر , كما يمكنك إلغاء هذا الشريط بعد ذلك باختيار أمر Delete Tab من هذه القائمة , أو يمكنك إعادة تسمية الشريط باختيار أمر Rename Tabe و يمكنك أيضاً ترتيب الأشرطة في الصندوق باختيار أمر Move Up و أمر Move Down

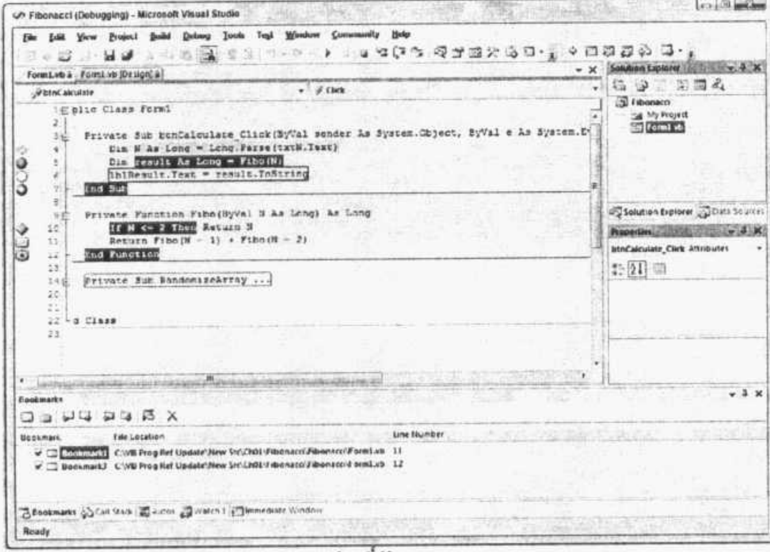
و بالنقر بزر الماوس الأيمن فوق أي من الأدوات الموجودة في صندوق الأدوات في أي شريط من الأشرطة التي يحتوي عليها هذا الصندوق تظهر قائمة مختصرة تحتوي علي عدة أوامر مثل أمر Cut و Copy و Past و التي تستخدم في نسخ و نقل الأدوات من شريط إلي أخر , بالإضافة إلي أمر Rename الذي يستخدم في إعادة تسمية الأدوات .

محرر الأكواد

يحتوي فيجوال أستوديو علي مجموعة من المحررات لمختلف أنواع المستندات بما في ذلك مجموعة من مختلف محررات الأكواد لكتابة الأنواع المختلفة من الأكواد بلغات البرمجة المختلفة التي يدعمها البرنامج مثل لغة TML و لغة XML و غيرها من اللغات , و تشترك هذه المحررات

مع بعضها البعض في العديد من الخواص الأساسية مثل خاصية إظهار التعليقات و الكلمات المفتاحية للغة بألوان مختلفة .

و من المحررات الموجودة في البرنامج محرر أكواد فيجوال بيسيك Visual Basic Code Editor الموضح في (شكل رقم ٣-٤) , و يحتوي هذا المحرر علي أرقام للأسطر لسهولة التعرف علي مكان الأجزاء المختلفة للأسكريبت , و لسهولة التعرف علي مكان الأخطاء في حالة حدوثها , و لإظهار أرقام الأسطر في محرر الأكواد أختار أمر Options من قائمة Tools و منها اختار Text Editor و منها General Page و قم بالنقر فوق Line Numbers



محرر الأكواد
(شكل رقم ٣-٤)

و في الهامش الأيسر لمحرر الأكواد تظهر مجموعة من الأيقونات هذه

الأيقونات تعطي معلومات عن السطر المقابل لكل أيقونة , فعلي سبيل المثال الأيقونة التي تأخذ شكل سهم أصفر تشير إلي أن تنفيذ البرنامج يتوقف عند هذا السطر , و الدائرة الحمراء تشير إلي وجود نقطة توقف في هذا السطر , و الدائرة الحمراء المفرغة تشير وجود نقطة توقف و لكن هذه النقطة غير منشطة , و الدائرة الحمراء الموجود بها علامة + تشير إلي نقطة توقف مشروطة , و الدائرة الحمراء المعتمة تشير إلي وجود نقطة توقف نشطة , و الأزرق و الأبيض يعني وجود إشارة مرجعية Bookmark

و من الممكن أن يوجد في الهامش الموجود في يسار السطر البرمجي أكثر من علامة من هذه العلامة إذا كان هذا السطر يحتوي علي أكثر من دلالة من الدلالات التي تشير إليها هذه العلامات .

القائمة المختصرة

و في الهامش الرمادي الموجود بيسار محرر الأكواد بالنقر بزر الماوس الأيمن فوق أيقونة نقطة التوقف تظهر القائمة المختصرة التي تحتوي علي مجموعة من الأوامر التي تستخدم في التعامل مع نقاط التوقف , فباختيار أمر Delete Breakpoint من هذه القائمة يمكنك إلغاء نقطة التوقف , و باختيار أمر Disable Breakpoint يمكنك تعطيل نقطة التوقف , و باختيار أمر Location من هذه القائمة يمكنك تغيير رقم السطر البرمجي الموجود فيه نقطة التوقف , و باختيار أمر Condition من هذه القائمة يمكنك وضع شرط لتوقف البرنامج عند نقطة التوقف , و باختيار أمر Hit Cout من هذه

القائمة يمكنك تحديد رقم عداد نقاط التوقف التي يتوقف عندها البرنامج عن التنفيذ , و باختيار أمر Filter من هذه القائمة يمكنك تقييد نقطة التوقف بحيث لا يتوقف البرنامج عند هذه النقطة إلا بتحقق شرط معين , و باختيار أمر When Hit من القائمة يمكنك تحديد الحدث الذي يحدث عندما يقوم البرنامج بالمرور علي هذه النقطة .

ألوان الأكواد

يعرض محرر الأكواد ألوان مختلفة للأكواد علي حسب نوع هذه الأكواد , و يمكنك تغيير الألوان الافتراضية التي يظهر بها المحرر كل نوع من أنواع الأكواد , فالتعليقات تظهر باللون الأخضر , و الأخطاء في قواعد اللغة تظهر تحتها خط أزرق , و الكلمات المفتاحية تظهر باللون الأزرق , و الأخطاء الأخرى تظهر باللون البنفسجي , و العبارات التي للقراءة فقط تظهر بأسفلها خلفية رمادية , و رسائل التحذير يظهر تحتها خط أخضر .

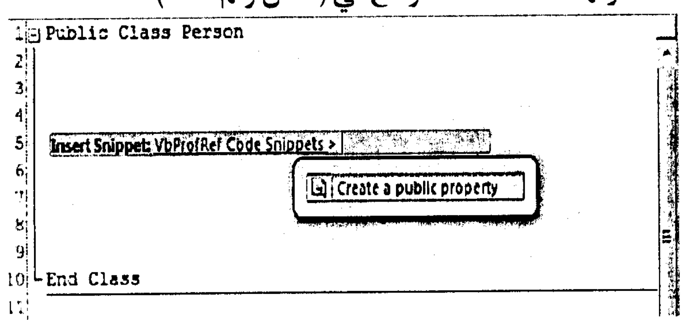
قصاصات الأكواد

قصاصات الأكواد Code Snippets عبارة عن مكتبة تحتوي علي قصاصات صغيرة أي ملفات صغيرة تحتوي علي أكواد هذه الأكواد يمكنك استخدامها في المشروع الذي تقوم بتنفيذه عن طريق أدراج هذه الأكواد في المشروع عند الحاجة إليها .

و يحتوي فيجوال أستوديو علي المئات من هذه القصاصات التي يمكنها أن تقوم بأداء أعمال نمطية في المشروعات , و تتميز هذه القصاصات بأنها سهلة

الاستعمال و سهلة الأدراج في المشروعات .

و لأدارج أي من هذه القصاصات أنقر بزر الماوس الأيمن في المكان الذي تريد أي من هذه القصاصات فيه و اختار Insert Snippet من القائمة المختصرة التي تظهر , ليقوم البرنامج بإظهار قائمة تضم مجموعات من هذه القصاصات , قم بالنقر المزدوج علي المجموعة التي تريدها ليتم عرض كل القصاصات الموجودة في المجموعة و باختيار أي من هذه القصاصات يظهر شرح مختصر لهذه القصاصة كالموضح في (شكل رقم ٣-٥) .



الشرح المختصر الذي يظهر لقصاصة الأكواد
(شكل رقم ٣-٥)

و بالنقر المزدوج علي القصاصة يتم أدراجها في المكان الذي قمت فيه بالنقر بزر الماوس الأيمن , و يمكنك إنشاء قصاصات الأكواد الخاصة بك لتخزين الأكواد التي تستخدمها باستمرار , و ذلك عن طريق أنشاء ملف XML لتعريف هذه الأكواد , و باستخدام أمر Snippets Manager و الضغط علي زر Add يتم فتح المتصفح الخاص بالملف الذي يحتوي علي القصاصة التي تريد الاحتفاظ بها .





5

أدوات التحكم

إنشاء أدوات التحكم
خواص أدوات التحكم
Properties
الأحداث
Events
هيكل بناء المشروع



إنشاء أدوات التحكم

أداة التحكم Control هي عبارة عن كائن برمجي Programing Object يتم وضعه في النموذج ليتفاعل مع أداء المستخدم , و من أمثلة أدوات التحكم التي توجد في النماذج الأزرار و الحقول النصية و أشرطة التمرير و قوائم الخيارات المنسدلة التي تتيح مع المستخدم التعامل مع البرنامج , و يتم تمثيل أدوات التحكم هذه في النموذج باستخدام أيقونات حتى تسهل علي المستخدم التعامل مع هذه الأدوات .

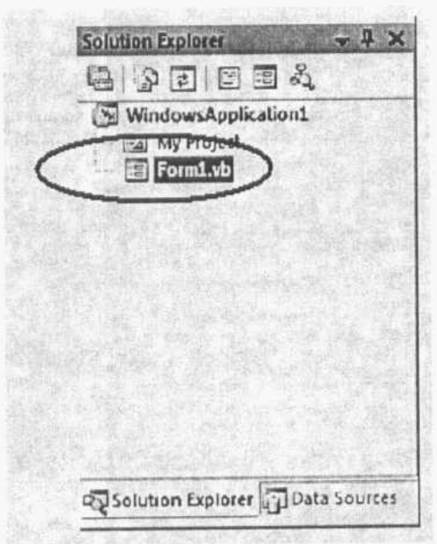
أنشاء أدوات التحكم

في الغالب يتم إضافة أدوات التحكم إلي النموذج أثناء إنشاء النموذج , كما أنك تستطيع أيضاً إضافة أدوات تحكم جديدة إلي النموذج أثناء تنفيذ المشروع و هذا مما يوفر لك نوع من المرونة في إنشاء المشروع حيث يمكنك تغيير شكل المشروع أثناء تشغيله وفقاً لمتطلبات التشغيل , و فيما يلي سنوضح كيفية أنشاء أدوات التحكم سواء عند تنفيذ المشروع أو عند تشغيل المشروع .

أنشاء أدوات التحكم عند تنفيذ المشروع

لإنشاء أدوات تحكم أثناء تنفيذ المشروع أنقر نقراً مزدوجاً فوق أيقونة النموذج Form في نافذة Solution Explorer الموجودة في يمين واجهة البرنامج كما يوضح (شكل رقم ٤-١) , لكي يتم فتح النموذج في نافذة Form Editor كما يوضح (شكل رقم ٤-٢) , و من شريط الأدوات

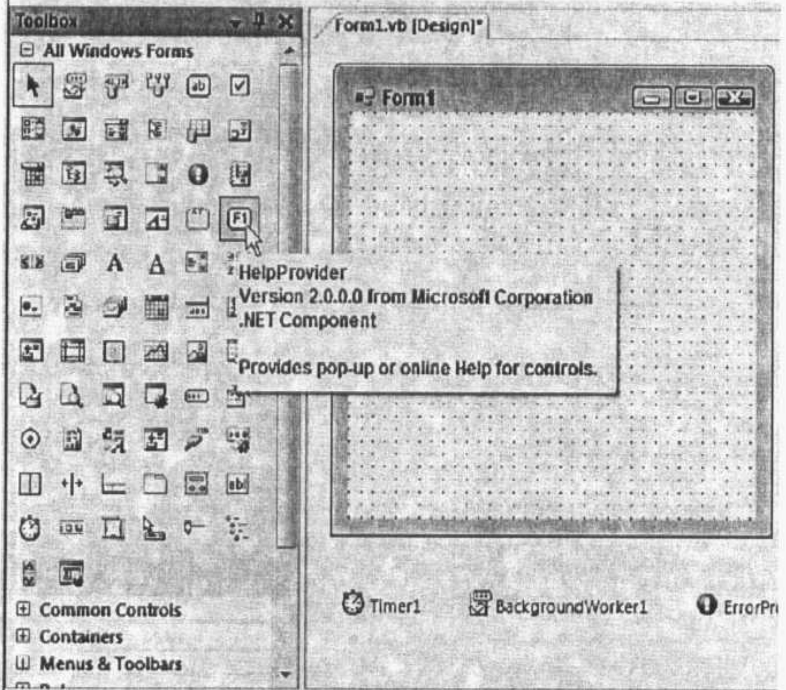
الموجود في يسار واجهة البرنامج كما يوضح (شكل رقم ٤-٢) قم باختيار أداة التحكم التي تريد إضافتها للنموذج , و بإيقاف مؤشر الماوس علي أي من أدوات التحكم الموجودة في شريط الأدوات يظهر شرحاً مختصراً لأداة التحكم بجوار مؤشر الماوس كما يوضح (شكل رقم ٤-٢) .



النقر علي أيقونة النموذج

(شكل رقم ٤-١)





فتح النموذج في نافذة Form Editor

(شكل رقم ٤-٣)

و بعد اختيار أداة التحكم التي تريد إضافتها للنموذج من شريط الأدوات يوجد عناك طرق عديدة لإضافة هذه الأداة إلي النموذج , فيمكنك إضافة الأداة إلي النموذج بالنقر عليها نقراً مزدوجاً في شريط الأدوات الموجودة به الأداة , و عند وضع أداة التحكم في النموذج بهذه الطريقة يتم وضعها بالحجم الافتراضي و في المكان الافتراضي في النموذج , كما يمكنك أيضاً إضافة



أدوات التحكم إلي النموذج عن طريق سحبها بزر الماوس من صندوق الأدوات الموجودة فيه أداة التحكم إلي النموذج المفتوح في واجهة البرنامج و في هذه الحالة أيضاً يتم إضافة الأداة إلي النموذج بالحجم الافتراضي و في المكان الافتراضي لها في النموذج , كما يمكنك أيضاً إضافة أكثر من نسخة لنفس الأداة إلي النموذج , و ذلك بالضغط علي مفتاح Ctrl أثناء سحب الأداة من صندوق الأدوات إلي النموذج في هذه الحالة تظل الأداة محددة في صندوق الأدوات حتى بعد إضافتها إلي النموذج , و بعد ذلك يمكنك إضافة نفس الأداة مرة أخري إلي النموذج ليس باستخدام السحب من صندوق الأدوات و إنما بمجرد النقر في النموذج في المكان الذي تريد إضافة الأداة فيه مرة أخري , و بعد الانتهاء من إضافة نسخ الأدوات إلي النموذج بهذه الطريقة قم بالنقر علي أداة السهم للخروج من هذه الخاصية .

كما يمكنك نسخ أدوات التحكم في النموذج باستخدام اختصارات لوحة المفاتيح الخاصة بذلك و هي Ctrl+C للنسخ و Ctrl+V للصق , و بنفس الطريقة أيضاً يمكنك نسخ أدوات التحكم من نموذج لأخر , و بذلك يكون من السهل عمل عدة نسخ من أدوات التحكم في النماذج لها نفس الخواص و لكن بأسماء مختلفة عن بعضها البعض , كما تسهل هذه الطريقة أيضاً في عمل مصفوفة من أدوات التحكم في النموذج , و لكن عندما تقوم بنسخ أدوات التحكم باستخدام اختصارات لوحة المفاتيح Ctrl+C و Ctr+V ففي هذه الحالة يقوم البرنامج بوضع كل النسخ التي تقوم بوضعها في النموذج في مكان واحد و

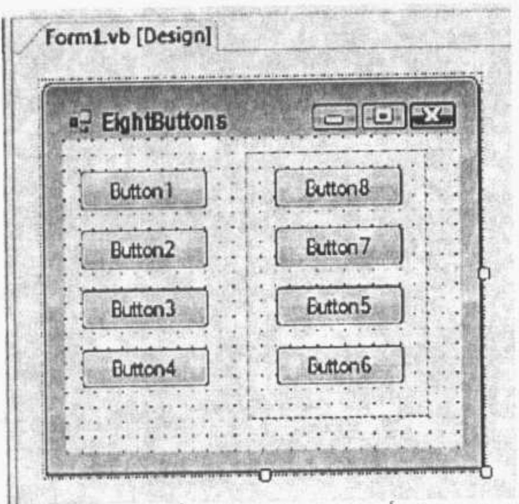
تحتاج بعد ذلك إلي إعادة ترتيب هذه الأدوات بالطريقة التي تريدها في النموذج .

أدوات التحكم

يمكنك تحديد أي من أدوات التحكم في داخل النموذج بمجرد النقر فوق هذه الأداة بزر الماوس , كما يمكنك تحديد أكثر من أداة تحكم في داخل النموذج عن طريق النقر فوق هذه الأدوات الواحدة تلو الأخرى مع الضغط علي مفتاح Ctrl في لوحة المفاتيح لإضافة أدوات جديدة إلي التحديد , أو بالنقر علي الأدوات المحددة بالفعل مع الضغط علي مفتاح Ctrl في لوحة المفاتيح لطرح هذه الأدوات من التحديد , كما يمكنك أيضاً تحديد مجموعة أدوات تحكم متجاورة في النموذج بالسحب بزر الموس حول هذه المجموعة المتجاورة من الأدوات لتحديدها معاً في وقت واحد , كما يمكنك بالضغط علي مفتاح Ctrl في لوحة المفاتيح أن تقوم بالسحب حول مجموعة أخري من أدوات التحكم بداخل النموذج مع الحفاظ علي التحديد الموجود فعلاً في النموذج .

و قد يحدث في بعض الأحيان أن تحتوي أدوات التحكم علي أدوات تحكم أخري كما يوضح (شكل رقم ٤-٣) .





احتواء بعض أدوات التحكم علي أدوات تحكم أخري

(شكل رقم ٤-٣)

و في هذه الحالة عندما تقوم بالسحب فوق الأداة التي تحتوي علي أداة لتحديدها يقوم فيجوال أستوديو بتحديد الأدوات الداخلية الموجودة بداخل الأدوات فقط ولا يقوم بتحديد الأدوات الأصلية التي تحتوي علي هذه الأدوات , و في هذه الحالة لتحديد الأدوات الداخلية و الأدوات الخارجية التي تحتوي عليها في نفس الوقت يجب عليك الضغط علي مفتاح Ctrl في لوحة المفاتيح أثناء السحب بالماوس حول الأدوات التي تقوم بتحديدها , كما يمكنك لو قمت بتحديد النموذج نفسه الذي يحتوي علي الأدوات و الضغط علي مفتاحي Ctrl+A في لوحة المفاتيح يمكنك في هذه الحالة تحديد كل أدوات التحكم التي يحتوي عليها النموذج .

و بعد تحديد أدوات التحكم الموجودة في داخل النموذج يمكنك التعامل مع هذه الأدوات حيث تكون محل لتطبيق الأوامر التي تختارها عليها , فيمكنك مثلاً إلغاء هذه الأدوات أو نسخها أو نقلها من مكان لأخر أو يمكنك تغيير خواص هذه الأدوات .

إنشاء أدوات التحكم عند تشغيل المشروع

قد تحتاج في بعض الأحيان إضافة أدوات تحكم إضافية إلي النموذج أثناء تشغيل المشروع بالإضافة إلي أدوات التحكم التي قمت بإضافتها للنموذج أثناء تنفيذ المشروع , كأن تري مثلاً أن الحقول النصية التي قمت بوضعها في النموذج غير كافية عند تشغيل النموذج و أردت أن تقوم بإضافة حقول نصية إضافية أو أي أدوات تحكم أخري في النموذج أثناء تشغيل المشروع , و الكود التالي يوضح الطريقة التي يستخدمها فيجوال أستوديو في إضافة أدوات تحكم إلي النموذج .

```
• Dim lbl As New Label
• lbl.SetBounds(10, 50, 100, 25)
• lbl.Text = "Hello World!"
• Me.Controls.Add(lbl)
```

و يمكنك إضافة أدوات تحكم إلي النموذج أثناء تشغيل المشروع عن طريق قيامك بفتح النموذج في محرر الأكواد ثم تقوم بتعريف المتغير الخاص بأداة التحكم , ثم تقوم باختيار أسم هذا المتغير من القائمة المنسدلة الموجودة في يسار المحرر ثم تقوم باختيار الحدث Event الذي يحدث عند استخدام أداة التحكم من قائمة الأحداث الموجودة في يمين محرر الأكواد و الكود



التالي يعتبر مثلاً لإضافة أوامر تحكم إلي النموذج أثناء تشغيل المشروع .

• 'تعريف المتغير الخاص بأداة التحكم.

```
• Private WithEvents btnHi As Button
```

• 'إنشاء الزر الخاص بأداة التحكم.

```
• Private Sub btnMakeHiButton_Click(ByVal sender As System.Object, _
• ByVal e As System.EventArgs) Handles btnMakeHiButton.Click
• btnHi = New Button
• btnHi.SetBounds(96, 50, 75, 23)
• btnHi.Text = "Say Hi"
• Me.Controls.Add(btnHi)
• End Sub
```

' • الحدث المرتبط بقيام المستخدم بالضغط علي الزر.

```
• Private Sub btnHi_Click(ByVal sender As Object, _
• ByVal e As System.EventArgs) Handles btnHi.Click
• MessageBox.Show("Hi")
• End Sub
```

كما يمكنك إلغاء أداة التحكم أثناء قيامك بتشغيل المشروع باستخدام الكود التالي :

```
• Me.Controls.Remove(btnHi)
• btnHi = Nothing
```

و الكود السابق يفترض أن أسم أداة التحكم هو btnHi

خواص أدوات التحكم Properties

خواص أدوات التحكم هي القيم المرتبطة بهذه الأدوات , فعلي سبيل المثال لو أن هناك أحد أدوات التحكم عبارة عن حقل نصي , فإن خواص هذا الحقل النصي تتمثل في لون خلفية النص BackColor و إحداثيات الحقل النصي



بالنسبة للركن الأيسر العلوي Top و Left تمثل خواص الحقل النصي و التي تمثل مكان الحقل في النموذج .

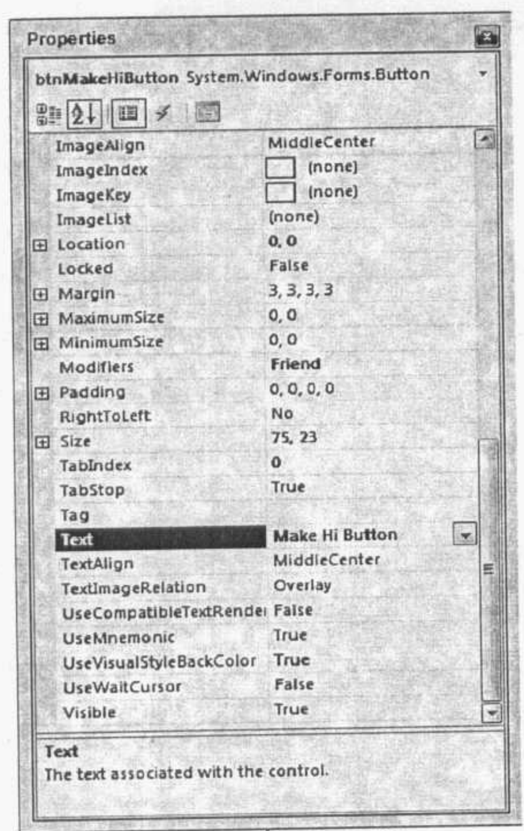
و كما في حالة إنشاء أدوات التحكم كما رأينا في الجزء السابق من هذا الفصل أنه يمكنك إنشاء أدوات التحكم بطريقة مرئية عند إنشاء المشروع و يمكنك إنشاء هذه الأدوات باستخدام الأكواد أثناء تشغيل المشروع , فإن تحديد خواص أدوات التحكم أيضا يتم بصورة مرئية أثناء إنشاء المشروع أو باستخدام الأكواد أثناء تشغيل المشروع , و فيما يل سوف نعرض الطريقة التي يتم تحديد خواص أدوات التحكم بها سواء أثناء تنفيذ المشروع أو أثناء تشغيل المشروع .

تحديد خواص أدوات التحكم أثناء تنفيذ المشروع

لتحديد خواص أدوات التحكم أثناء تنفيذ المشروع قم بفتح النموذج الذي يحتوي علي هذه الأدوات في نافذة تصميم النموذج Form Designer و قم بالنقر فوق أداة التحكم في النموذج التي تريد تحديد خواصها لتظهر بذلك نافذة خواص أدوات التحكم كما يوضحها (شكل رقم ٤-٤) , و توضح هذه النافذة خواص أدوات التحكم كالأزرار و الحقول النصية .







نافذة خواص أدوات التحكم
(شكل رقم ٤-٤)

و تعرض قائمة btnMakeHiBtton الموجودة في أعلي نافذة الخواص تعرض أسماء الخواص التي يمكن إضافتها لعنصر التحكم المحدد في النموذج , و يمكنك تحديد قيم هذه الخواص بمجرد النقر فوق أي من هذه القيم في النافذة , فمن خلال القيم الموجودة في هذه القائمة يمكنك تحديد الحد

الأدنى و الحد الأعلى و القيمة الحالية لكل خاصية من الخواص التي تقوم بإضافتها لعنصر التحكم المحدد في النموذج .

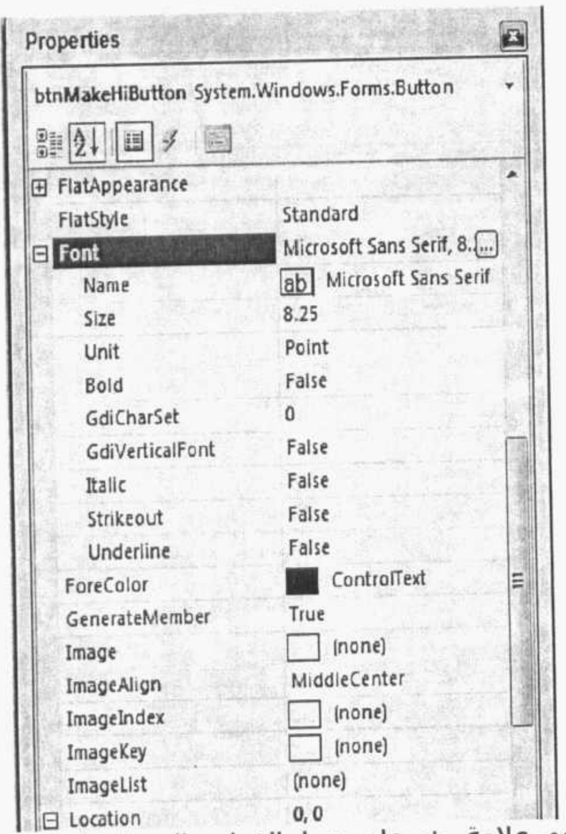
الخواص المركبة

الخواص المركبة Comound Properties هي الخواص التي لها قيم مركبة مثل الخاصية التي تقوم بتحديد مكان عنصر التحكم في داخل النموذج بالنسبة للركن الأيسر العلوي من النموذج و هي الخاصية Location هذه الخاصية خاصية مركبة لأنها تحتوي علي القيمتين X و Y واللتان تحددان المسافة بين عنصر التحكم و بين الحافة العلوية و الحافة أليسري للنموذج الموجود فيه عنصر التحكم , و كذلك خاصية الحجم Size و التي تحدد حجم عنصر التحكم فهذه الخاصية خاصية مركبة أيضاً لاحتوائها علي قيم للعرض Width والارتفاع Height الخاص بعنصر التحكم .

و تظهر الخواص المركبة أو الخاص التي لها قيم مركبة تحتوي علي أكثر من قيمة في نافذة الخواص و بجوارها إلي اليسار علامة + كالموضحة في (شكل رقم ٤-٥) , و بالنقر علي هذه العلامة تظهر القيم المتعددة المتعلقة بهذه الخاصية .

و بالنقر علي علامة + الموجودة بجوار الخواص التي لها قيم مركبة تتحول هذه العلامة إلي علامة - و تظهر كل القيم المخفية خلف علامة + و يمكنك اختيار أي من هذه القيم بالنقر عليها في القائمة .



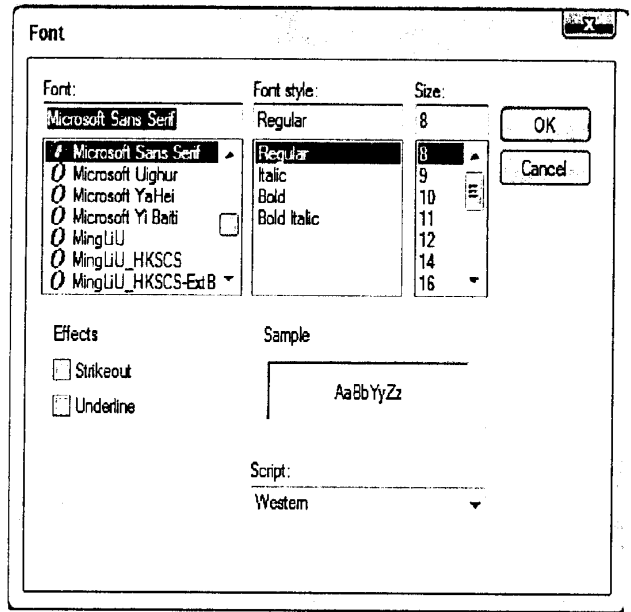


ظهور علامة + علي يسار الخواص التي لها قيم مركبة
(شكل رقم ٤-٥)

و في بعض الأحيان يمكنك تحديد كل قيم الخواص المركبة في خطوة واحدة بالنقر علي الزر الموجود في يمين هذه الخاصية في نافذة الخواص , كالنقر مثلاً علي هذا الزر الموجود بجوار خاصية Font و الذي يظهر المربع الحواري الموضح في (شكل رقم ٤-٦) و الذي من خلاله يمكنك تحديد كل



قيم خواص الخط من حيث الحجم و التنسيق و نوع الخط .



المربع الحواري الخاص بتحديد كل قيم خواص الخط

(شكل رقم ٤-٦)

الخواص المقيدة

الخواص المقيدة هي الخواص التي لا تحتوي إلا علي عدد قليل من القيم كقيمتين فقط لا غير مثل القيم المنطقية Boolean Values و التي لا يوجد لها سوي قيمتين فقط القيمة True و القيمة False كخاصية الإخفاء و الإظهار مثلاً Visible لها قيمتين فقط هي أما أن يكون عنصر التحكم ظاهراً True أو مخفي False و يوجد بجوار مثل هذه الخواص في نافذة الخواص



سهم صغير بالنقر علي هذا السهم تظهر قائمة تحتوي علي القيمتين المتعلقتين بهذه الخاصية القيمة True و القيمة False و من هذه القائمة يمكنك اختيار أي من هاتين القيمتين للخاصية .

و هناك نوع أخر من الخواص المقيدة تحتوي علي عدد أخر من القيم كالخاصية Flatstyle التي يتم إضافتها لأزار النموذج و التي تحدد شكل هذا الزر , فهذه الخاصية لها أربعة قيم محددة تظهر في القائمة التي تظهر بالنقر علي الزر الأسود الموجود بيمين الخاصية في نافذة الخواص كما يوضحها (شكل رقم ٤-٧) , و التي تمكنك من اختيار أي من هذه القيم الأربعة كقيمة للخاصية بالنقر عليها في القائمة .



اختيار أحد خواص القيم المقيدة من القائمة

(شكل رقم ٤-٧)

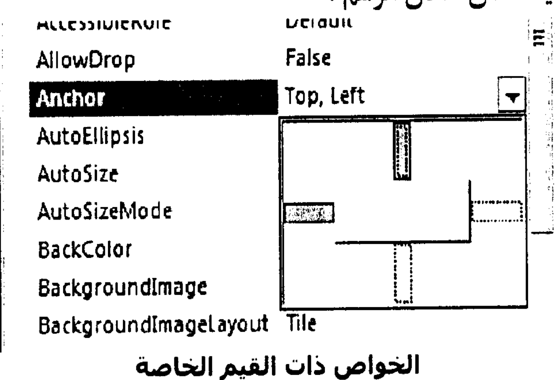
كما يمكنك أيضاً التنقل بين القيم المختلفة للخاصية بالنقر عليها بزر الماوس في نافذة الخواص بدلاً من اختيار القيم من القائمة الفرعية التي تظهر





بالنقر علي السهم الأسود الموجود بيمين الخاصية .

كما يوجد بعض الخواص التي لها قيم خاصة مثل الخاصية Anchor و التي تتمثل قيمها في حجمها بالنسبة للاتجاهات الأربعة , فباختيار هذه الخاصية من قائمة الخواص و النقر علي السهم الأسود الصغير الموجود بيمين هذه الخاصية يظهر الرسم الموضح في (شكل رقم ٤-٨) , و الذي من خلاله تستطيع تحديد قيم هذه الخاصية بالنسبة للاتجاهات الأربعة الموضحة في الرسم , و بعد الانتهاء قم بالضغط علي مفتاح Enter في لوحة المفاتيح لقبول القيم التي قمت بتحديدها من خلال الرسم .



الخواص ذات القيم الخاصة
(شكل رقم ٤-٨)

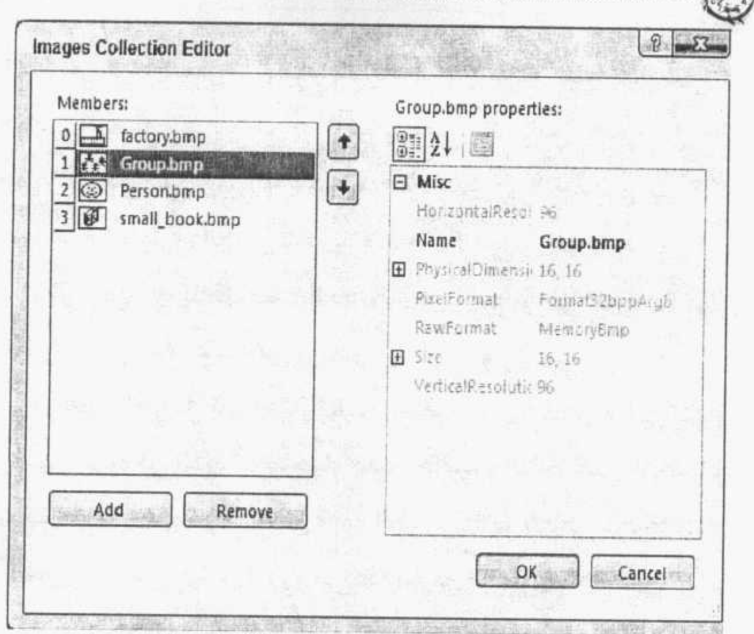
الخواص المجمعة

هناك بعض الخواص المجمعة معاً و التي يتم إعطاؤها قيم جماعية فعلي سبيل المثال مجموعة خواص ListBok و التي تحتوي علي قائمة من الخواص هذه الخواص يتم تحديد قيمها معاً في وقت واحد لأن قيم هذه الخواص تكون



مرتبطة مع بعضها البعض , و تقوم قائمة الخواص بعرض مثل هذا النوع من الخواص كمجمع Collection و باختيار مجمع الخواص هذا المن قائمة الخواص و النقر علي الزر الدائري الموجود بيمين هذا المجمع في القائمة يظهر المربع الحواري الخاص بتحديد قيم هذه الخواص كمجموعة معاً كما, و بعد الانتهاء من تحديد قيم هذه المجموعة من الخواص في المربع الحواري أضغط زر Ok لحفظ هذه القيم و إغلاق المربع الحواري .

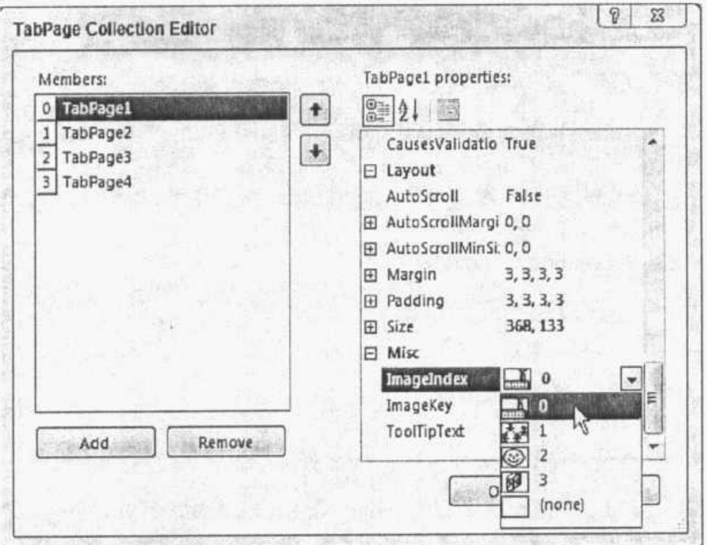
و بعض الخواص تكون أكثر تعقيداً كخاصية ImageList و التي تستخدم في إظهار صورة في الشريط , فباختيار هذه الخاصية و الضغط علي الزر الدائري الموجود بيمين هذه الخاصية في قائمة الخواص يظهر بذلك المربع الحواري الموضح في (شكل رقم ٤-٩) , و في هذا المربع الحواري بالنقر علي زر Add يظهر المربع الحواري الخاص بإضافة الصورة إلي عنصر التحكم , و القائمة الموجودة في اليسار توضح معاينة لكل قيمة من القيم التي تقوم بتحديها للصورة التي تقوم بإضافتها إلي عنصر التحكم .



المربع الحواري الخاص بتحديد قيم الخواص المجمعة
(شكل رقم ٤-٩)

و بعد إضافة الصورة إلي خاصية ImageList يتم إنشاء خاصية TabControl و باختيار هذه الخاصية من قائمة الخواص و الضغط علي الزر المستدير الموجود بيمين هذه الخاصية في القائمة يظهر المربع الحواري الموضح في (شكل رقم ٤-١٠) الخاص بتحديد خواص هذه الخاصية .





المربع الحواري الخاص بتحديد قيم خاصية TabControl

(شكل رقم ٤-١٠)

تحديد قيم الخواص أثناء تشغيل البرنامج

يمكنك كتابة كود لتحديد قيم الخواص المتعلقة بأدوات التحكم , و ذلك بكتابة أسم أداة التحكم , و كتابة نقطة بعدها , ثم كتابة أسم الخاصية بعد هذه النقطة , ثم إضافة قيمة الخاصية بعد علامة التساوي = .

فعلي سبي المثال الكود التالي يقوم باختبار النص Text الموجود في الحقل النصي TextBok الذي أسمه TextPath فإذا كان هذا النص لا ينتهي بشرطة مائلة / يقوم الكود بإضافة هذه الشرطة إلي نهاية النص , و بذلك فهذا الكود يقوم بقراءة و تحديد هذه الخاصية في النص .



```
• If Not txtPath.Text.EndsWith("/") Then txtPath.Text &= "/"
```

و الكود التالي يقوم بإظهار رسالة توضح ما إذا كان النص TxtPath مكتوب بخط عريض Bold فإذا لم يكن كذلك يقوم الكود بكتابته بالخط العريض , أي تغيير قيمة خاصية تنسيق النص Font إلي الخط العريض .

```
• If txtPath.Font.Bold Then
• MessageBox.Show("Bold")
• Else
• MessageBox.Show("Not Bold")
• End If
```

الأداء

المقصود بالأداء Method هو تنفيذ الكود المرتبط بأداة التحكم , و قد يكون أداء أداة التحكم عبارة عن دالة يتم تنفيذها هذه الدالة تعود بقيمة معينه , أو قد يكون أداء أداة التحكم عبارة عن تنفيذ برنامج كامل .

الأحداث Events

الحدث Event هو الفعل الواجب حدوثه حتى يتم تنفيذ الكود أو الدالة أو البرنامج الخاص بأداة التحكم , فلو كانت أداة التحكم هي عبارة عن أحد أزرار النموذج , فإن الحدث الخاص بتشغيل هذه الأداة هو النقر بزر الماوس فوق هذا الزر , فعملية النقر برز الماوس علي أداة التحكم و هو الزر الموجود في النموذج تسمي في عالم البرمجة حدث Event و تختلف الأحداث باختلاف أداة التحكم , فقد يكون الحدث عبارة عن النقر بزر الماوس فوق أداة التحكم , أو قد يكون الحدث مجرد مرور مؤشر الماوس فوق أداة التحكم , أو قد يكون الحدث اللازم لتشغيل أداة التحكم هو الضغط علي أحد أو بعض المفاتيح في

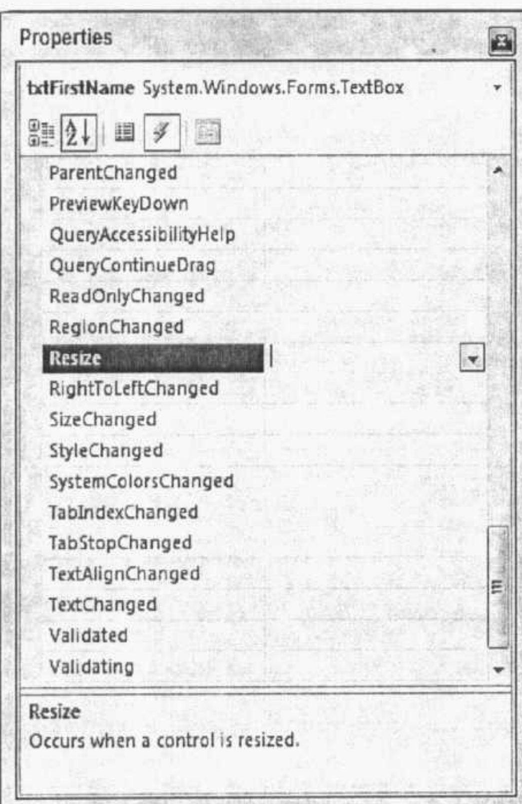
لوحة المفاتيح .

تحديد الأحداث

لو قمت بفتح النموذج في نافذة تصميم النموذج Form Designer و قمت بالنقر فوق احد أدوات التحكم الموجودة في النموذج نقراً مزدوجاً يتم فتح محرر الأكواد , و يظهر في محرر الأكواد الأحداث الافتراضية لتشغيل أداة التحكم , فالزر الحدث الافتراضي لتشغيله هو النقر علي هذا الزر بزر الماوس , و الحقل النصي الحدث اللازم لتشغيله هو أدراج نص في داخل هذا الحقل , و الحدث الخاص بتشغيل النموذج نفسه هو فتح هذا النموذج , و يمكنك قبول هذه الأحداث الافتراضية الخاصة بأدوات التحكم , أو إذا أردت يمكنك تغيير هذه الأحداث الافتراضية .

و لتغيير الأحداث الافتراضية اللازمة لتشغيل أحد أدوات التحكم , قم بتحديد أداة التحكم , ثم قم بالنقر فوق زر Events في نافذة خواص أدوات التحكم Properties Window لتظهر بذلك قائمة الأحداث البديلة التي يمكنك استخدامها بدلاً من الحدث الافتراضي لأداة التحكم كما يوضحها (شكل رقم ٤ -١١)

و يتم اختيار الحدث المصاحب لأداة التحكم بالنقر عليه في هذه القائمة نقراً مزدوجاً .

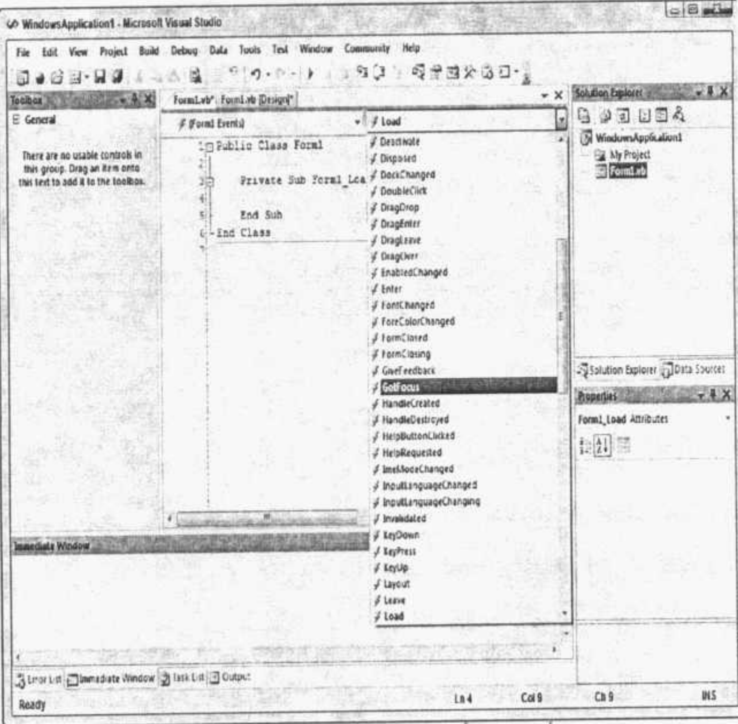


قائمة الأحداث الممكنة لتشغيل أداة التحكم
(شكل رقم ٤-١١)

كما يمكنك أيضاً استخدام محرر الأكواد في تخصيص حدث معين لأداة تحكم معينه , و ذلك بفتح محرر الأكواد Code Editor و اختيار أداة التحكم التي تريد إضافة حدث لها من قائمة أدوات التحكم الموجودة بيسار نافذة محرر الأكواد , ثم اختار الحدث الذي تريد أن يصاحب أداة التحكم من قائمة الأحداث



الموجودة في اليمين كما يوضح (شكل رقم ٤-١٢) .



تخصيص الأحداث لأدوات التحكم باستخدام محرر الأكواد
(شكل رقم ٤-١٢)

هيكل بناء المشروع

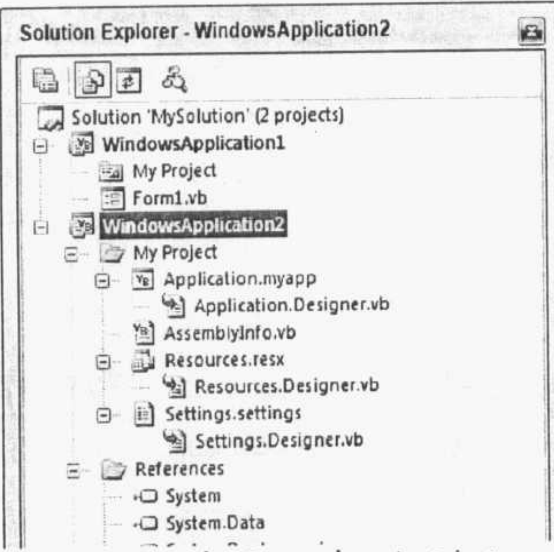
في فيجوال أستوديو يمكنك التعامل مع مجموعة من المشروعات تحتوي علي أكثر من مشروع في نفس الوقت هذه المشروعات تربطها معاً علاقة معينه , و هذه المجموعة Solution من المشروعات يمكن فتحها معاً في نفس الوقت في

البرنامج , و يتكون كل مشروع من هذه المشروعات علي مجموعة من الملفات و البرامج و المكتبات , و فيما يلي عرض لمجموعة الملفات التي يتكون منها المشروع .

(شكل رقم ٥-١) يوضح متصفح المشروعات Solution Explorer و المتصفح يحتوي علي مشروعين هما المشروع WindowsApplication1 و المشروع WindowApplication2 و كل مشروع من المشروعين يحتوي علي عناصر المشروع , و التي تتمثل في الملفات التي تحتوي علي أعدادات المشروع و النماذج و التي تسمي Form1

و في الشكل لاحظ الفارق بين الطريقة التي تعرض بها ملفات المشروع الأول WindowsApplication1 في المتصفح و بين الطريقة التي تعرض بها ملفات المشروع الثاني WindowsApplication2 و الفارق هنا هو أن الملفات المخفية غير ظاهرة في المشروع الأول بينما هذه الملفات تظهر في المتصفح في المشروع الثاني , و ذلك بسبب النقر علي الزر الخاص بإظهار الملفات المخفية في المتصفح و الموجودة بجوار أسم المشروع , أي أنه بالنقر علي زر إظهار الملفات المخفية يظهر هيكل بناء المشروع في المتصفح و يعرض كل الملفات التي يتكون منها المشروع .





متصفح مشروعات يحتوي علي مشروعين

(شكل رقم ٥-١)

و تقوم فيجوال بيسيك بإنشاء هذه المجموعة من المجلدات و الملفات لأغراض مختلفة فكل ملف من هذه الملفات له وظيفة معينه يقوم بأدائها , و الجدول التالي يوضح وظيفة كل ملف من هذه الملفات .

أسم المجلد أو الملف	وظيفته
WindowsApplication2	يحتوي هذا المجلد كل المشروع
My Project	يحتوي هذا المجلد علي كل المعلومات المتعلقة بالمشروع , و التطبيقات الخاصة به و



أسم المجلد أو الملف	وظيفته
	الأحداث و الأعدادات , و بالنقر بزر الماوس نقراً مزدوجاً علي هذا المجلد يمكنك تعديل القيم التي يحتوي عليها
Application.myapp	عبارة عن ملف XML يقوم بتعريف خواص التطبيقات
Application.designer.vb	يحتوي هذا الملف علي الأكواد التي تعمل مع القيم المحدد في ملف Application.myapp
AssemblyInfo.vb	يحتوي هذا الملف علي المعلومات المتعلقة بالتطبيقات مثل معلومات حقوق الطبع و أسم الشركة و العلامة التجارية و الإصدار
Resources.resx	يحتوي هذا الملف علي الأكواد الأصلية للمشروع
Resources.Designer.vb	يحتوي هذا الملف علي أكواد فيجوال بيسيك الخاصة بالتعامل مع الملفات التي تحتوي علي الأكواد الأصلية الموجودة في ملف Resources.resx
Settings.settings	يحتوي هذا الملف علي الأعدادات الخاصة بالتطبيقات

أسم المجل أو الملف	وظيفته
Settings.Designer.vb	يحتوي هذا الملف علي أكواد فيجوال بيسيك الخاصة بالتعامل مع الأعدادات الموجودة في ملف Settings.settins
References	هذا المجلد متعلق بالمكونات المستخدمة من خارج فيجوال بيسيك مثل ملفات DLL و ملفات COM
bin	يستخدم هذا المجلد في بناء البرامج قبل تنفيذها حيث يحتوي علي ملفات exe. التي تم تفسير الأكواد الموجودة بها
obj	يستخدم هذا المجلد في بناء البرامج قبل تنفيذها
ApplicationEvents.vb	هذا الملف يحتوي علي الأكواد الخاصة بالأحداث الخاصة بالبرامج
Form1.vb	هذا الملف خاص بالنموذج حيث يحتوي علي الأكواد المتعلقة بالنموذج التي تقوم بكتابتها و المتعلقة بأدوات التحكم في النموذج و الأحداث الخاصة بأوامر التحكم
Form1.Designer.vb	هذا الملف يحتوي علي أكواد فيجوال بيسيك



أسم المجلد أو الملف	وظيفته
	التي تم تحريرها بواسطة المصمم و الخاصة ببناء النموذج

جدول يوضح الملفات التي يتكون منها المشروع و وظيفة كل ملف

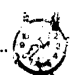
و قد لا تحتاج أثناء قيامك بالعمل في المشروع أن تقوم بالتعامل مع كل هذه الملفات.





قواعد لغة فيجوال بيسيك

- أنواع البيانات
- المتغيرات
- المعاملات Operators
- الجمل البرمجية Statements
- الفئات Classes



أنواع البيانات

تعتبر المتغيرات من أهم عناصر بناء البرامج , فالمتغير عبارة عن كائن برمجي يستخدم في تخزين قيمة معينه هي قيمة هذا المتغير , و تأخذ قيمة المتغير عدة أشكال , فقد تكون قيمة المتغير عبارة عن رقم أو عدد صحيح أو كثر عشري أو حرف أو كلمة مكونة من مجموعة من الأحرف أو قيمة منطقية مثل القيمة True أو القيمة False

و عندما يتم تعريف المتغير و نوع القيمة التي يمكن أن تعطي لهذا المتغير , و إعطاء قيمة لهذا المتغير فإن البرنامج يستطيع أن يتعامل مع هذه القيمة و أن يقوم بأجراء العمليات الحسابية عليها و البحث عنها و العمليات المنطقية و عمليات المقارنة لهذه القيمة .

و بوجه عام تتوقف الطريقة التي يتعامل بها المتغير علي أربعة خواص أساسية لهذا المتغير هذه الخواص الأربعة هي نوع القيمة التي يأخذها المتغير ما إذا كانت هذه القيمة عبارة عن عدد صحيح أو كسر عشري أو نص أو قيمة منطقية , و الخاصية الثانية هي نطاق المتغير و الخاصية الثالثة هي إمكانية الوصول إلي المتغير , و الخاصية الرابعة هي عمر المتغير نفسه .

نوع البيانات

الجدول التالي يوضح الأنواع المختلفة للبيانات و مثال علي كل نوع من هذه الأنواع .

نوع البيانات	مثال
قيمة منطقية	True أو False
عدد صحصح	133
عدد طويل	100000000
كسر عشري	79,228
قيمة مزدوجة	من 1.79769313486231570E+308l إلي -4.94065645841246544E-324
قيمة لفظي	من صفر إلي ٢ بليون حرف
تاريخ	من January 1, 0001 0:0:00 إلي December 31, 9999 11:59:59 pm
كائن	أي نوع من البيانات
هيكل	مجموعة مركبة من البيانات

جدول يوضح الأنواع المختلفة للبيانات

و الأعداد الصحيحة تعتبر أسرع أنواع القيم في تعامل البرنامج معها , فاستخدام هذا النوع من البيانات يؤدي إلي تسريع البرنامج الذي تقوم بتنفيذه

و تستخدم فيجوال بيسيك حروفاً معينه كرمز لتحديد كل نوع من أنواع البيانات , و الجدول التالي يوضح الرموز المختلفة المستخدمة في فيجوال بيسيك للدلالة علي كل نوع من أنواع البيانات .



نوع البيانات	الرمز الدال عليها
قيمة عددية صحيحة	%
عدد طويل	&
كسر عشري	@
قيمة مفرده	!
قيمة مزدوجة	#
قيمة لفظية	$

جدول يوضح الرموز الدالة علي أنواع البيانات

و تستخدم هذه الرموز في تحديد نوع البيانات التي تحتوي عليها المتغيرات عن طريق وضع الرمز علي يمين أسم المتغير عند تعريف هذا المتغير , و علي سبيل المثال الكود التالي يقوم بتعريف المتغير satisfaction_quotient كقيمة زوجية و ذلك بوضع علامة # علي يمين أسم المتغير .

- num_desserts = 100
- satisfaction_quotient# = 1.23

و عندما تقوم باستخدام المتغير بعد ذلك يقوم البرنامج بالتعامل مع هذا المتغير علي أساس النوع الذي قمت بتحديده له عند قيامك بتعريف المتغير في أول مرة , كما تقوم فيجوال بيسيك بتغيير نوع المتغير بعد أن تقوم بتحديد هذا النوع عند تعريفه في حالة إعطاء قيمة للمتغير تختلف في النوع مع النوع الذي قمت بتحديده للمتغير عند تعريفه في أول مرة , فعلي سبيل المثال لو قمت بتعريف متغير علي أنه عدد صحيح ثم قمت بإعطاء هذا المتغير قيمة كبيرة جداً

ففي هذه الحالة تقوم فيجوال بيسيك بتحويل نوع المتغير بشكل تلقائي إلي عدد طويل .

و في المثال التالي يوجد ثلاثة أسطر خاصة بإعطاء قيم لثلاثة متغيرات .

- an_integer = 100
- a_long = 10000000000
- a_double = 1.0

في السطر الأول تم إعطاء المتغير an_integer القيمة 100 و هي قيمة عدد صحيح , و في السطر الثاني تم إعطاء المتغير a_long القيمة 1000000000000 و هي قيمة عدد كبير , و في السطر الثالث تم إعطاء المتغير a_double قيمة مزدوجة هي 1.0 و تتعامل فيجوال بيسيك علي أنواع المتغيرات الثلاثة علي حسب القيم المعطاة لهذه المتغيرات , و لو قمت بإعطاء أي من هذه المتغيرات بعد ذلك القيمة True أو القيمة False مثلاً تقوم فيجوال بيسيك بتغيير نوع هذا المتغير إلي متغير منطقي بشكل تلقائي و التعامل مع بعد ذلك علي هذا الأساس .

و التواريخ في فيجوال بيسيك يتم التعبير عنها بعلامة الشباك # و لو قمت بتحديد قيمة في شكل تاريخ لأحد المتغيرات تقوم فيجوال بيسيك بالتعامل مع هذا المتغير علي أنه تاريخ كما في المثال التالي :

- a_boolean = True
- a_date = #12/31/2007#

و بالإضافة إلي العلامات التي توضع بعد اسم المتغير لتحديد نوع هذا المتغير يوجد في فيجوال بيسيك حروفاً لتدل علي نوع القيم التي يحتوي عليها هذا المتغير , و الجدول التالي يوضح الحروف الدالة علي الأنواع المختلفة



للبيانات التي يمكن إعطاؤها للمتغيرات .

نوع البيانات	الحرف الدال علي نوع البيانات
بيانات قصيرة	S
عدد صحيح	I
عدد طويل	L
كسر عشري	D
قيمة مفرده	F
قيمة زوجية	R

جدول يوضح الحروف التي تستخدم للدلالة علي الأنواع المختلفة لقيم المتغيرات

تغيير نوع البيانات

في الغالب يتم إعطاء قيمة للمتغير تتناسب مع نوع هذا المتغير الذي تم تحديده عند تعريف المتغير لأول مرة , فالمتغير اللفظي String Variable علي سبيل المثال يتم إعطاء قيمة لفظية له عبارة عن مجموعة من أحرف الهجاء , و المتغير العددي الصحيح Intger يتم إعطاؤه قيمة عبارة عن عدد صحيح , و لكن كما سبق و أن ذكرنا يمكنك إعطاء قيمة للمتغير تخالف نوع المتغير و تقوم فيجوال بيسيك تلقائياً بتغيير نوع المتغير علي حسب نوع القيمة المعطاة للمتغير .

و لكن عند قيامك بإعطاء قيمة للمتغير تخالف نوع هذا المتغير فإن المتغير

لا يستطيع استيعاب كل القيمة المعطاة له من نوع مخالف لنوع المتغير , فعلي سبيل المثال الكود التالي يقوم بنسخ القيمة من متغير طويل Long Varible إلي متغير عددي صحيح Intger Variable و من المعروف أن المتغير الطويل يستطيع أن يحتوي علي قيمة كبيرة جداً لا يستطيع أن يحتويها المتغير العددي العادي , علي ذلك لن يستطيع المتغير العددي احتواء كل القيمة .

- Dim an_integer As Integer
- Dim a_long As Long
- an_integer = a_long

و بنفس الطريقة عندما تقوم بإعطاء قيمة نصية String لمتغير عددي Intger كما في الكود التالي , ففي هذه الحالة لو كان النص الذي تقوم بإعطائه للمتغير العددي يحتوي علي عدد فإن المتغير العددي يستطيع قبول هذه القيمة , أما لو لم تحتوي القيمة التي تعطيها للمتغير العددي علي أعداد ففي هذه الحالة لن يقبل المتغير العددي القيمة النصية , و في هذه الحالة تظهر رسالة خطأ عند تنفيذ الكود .

- Dim an_integer As Integer
- Dim a_string As String
- an_integer = a_string

و الجدول التالي يوضح الدوال المستخدمة في فيجوال بيسيك لتغيير أنواع القيم .

الدالة	النوع الذي تؤدي إلي التغيير إليه
Cbool	قيمة منطقية
Cbyte	بايت



Cdat	تاريخ
CDbl	قيمة مزدوجة
Cdec	قيمة عشريه
Cint	عدد صحيح
CLng	قيمة عددية طويلة
Cobj	كائن
Schort	قيمة قصيرة
CSng	مفرد
CStr	قيمة لفظية

جدول يوضح الدوال المستخدمة لتحويل أنواع القيم

المتغيرات

سبق و أن ذكرنا أن المتغير عبارة عن مخزن للقيمة , و يتم تعريف المتغير Variable Declaration عن طريق تحديد أسم هذا المتغير و نوعه , و فيما يلي الصيغة العامة لتعريف المتغير في فيجوال بيسيك .

- [attribute_list] [accessibility] [Shared] [Shadows] [ReadOnly] _
- Dim [WithEvents] name [(bounds_list)] [As [New] type] [= initialization_expression]

و كما ينضح من الصيغة العامة المستخدمة في تعريف المتغير في فيجوال بيسيك أن تعريف المتغير يتضمن تحديد أسم المتغير و نوعه , و فيما يلي أمثلة للأكواد المستخدمة في تعريف المتغيرات في فيجوال بيسيك .

```
• Dim i = 1
• Public j As Integer
```

فالكود الأول يقوم بتعريف المتغير i علي أنه متغير عددي صحيح لإعطائه قيمة عددية صحيحة و هي 1 و الكود الثاني يعرف المتغير j بأنه متغير عددي صحيح و لكن بطريقة مختلفة و هي استخدام جمله As Integer

و الكود التالي يقوم بتحديد خوص ثلاثة متغيرات هي المتغير ItemName و المتغير Quantity و المتغير Price وفي الكود يتم تعريف المتغير ItemName علي أنه متغير لفظي حيث سيستخدم هذا المتغير في تحديد أسم المنتج , و المتغير Price تم تعريفه في الكود علي أنه قيمة عشرية حيث سيستخدم في تحديد سعر المنتج , و المتغير Quanitiy تم تعريفه علي أنه عدد صحيح حيث سيستخدم في تعريف الكمية المطلوبة من المنتج , و هذه المتغيرات الثلاثة موجودة ضمن فئة Class تسمي OrderItem

```
• Public Class OrderItem
• <XmlText()> _
• Public ItemName As String
• <XmlAttributeAttribute(AttributeName:="Cost")> _
• Public Price As Decimal
• <XmlAttributeAttribute()> _
• Public Quantity As Integer
• End Class
```

الكلمات المفتاحية المستخدمة عند تعريف المتغير

هناك العديد من الكلمات المفتاحية التي يمكن استخدامها في فيجوال بيسيك عند تعريف المتغير لتحديد خواص معينه للمتغير لتوفير قدر أكبر من



المعلومات حول المتغير لمفسر الأكواد .

و فيما يلي عرض للكلمات المفتاحية المستخدمة في تحديد خواص المتغير

الكلمة المفتاحية Puplic

استخدام الكلمة المفتاحية Puplic عند تعريف المتغير تفيد أن هذا المتغير متغير عام يمكن الوصول إليه سواء من داخل الأسكريبت الموجود فيه هذا المتغير أو من خارج هذا الأسكريبت .

الكلمة المفتاحية Protected

تفيد هذه الكلمة أن المتغير لا يمكن الوصول إليه إلا من خلال الأكواد الموجودة داخل الفئة الموجود فيها هذا المتغير فقط .

الكلمة المفتاحية Friend

تفيد الكلمة المفتاحية Friend التي تستخدم عند تعريف المتغير أن هذا المتغير يمكن الوصول إليه من داخل أو من خارج الأسكريبت الموجود فيه المتغير و لكن في نفس المشروع المستخدم فيه المتغير فقط أي لا يمكن الوصول إلي هذا المتغير من أي مشروع أخر .

الكلمة المفتاحية Private

تشير الكلمة المفتاحية Private التي تستخدم عند تعريف المتغير في أن هذا المتغير خاص بفئة معينه لا يمكن الوصول إليه من أي فئة أخري .

الكلمة المفتاحية Static

تفيد الكلمة المفتاحية Static التي تستخدم عند تعريف المتغير في أن قيمة

هذا المتغير صالحة فقط أثناء فترة حياة المتغير .

الكلمة المفتاحية Shared

هذه الكلمة المفتاحية تجعل المتغير الواحد مشترك بين أكثر من فئة و يمكن استخدامه في أي من هذه الفئات التي تشترك في هذا المتغير .

الكلمة المفتاحية Shadows

تجعل هذه الكلمة المفتاحية المتغير الذي ستخدم معه يقوم بإخفاء أي متغير أخر يحمل نفس أسم هذا المتغير و يحل محله .

الكلمة المفتاحية Read Only

هذه الكلمة تحمي المتغير من تغيير قيمته و تجعل المتغير للقراءة فقط بحيث لا يستطيع البرنامج تغيير قيمة هذا المتغير , و المثال التالي يوضح كيفية استخدام هذه الكلمة المفتاحية .

- Public Class EmployeeCollection
- Public ReadOnly MaxEmployees As Integer = 100
- End Class

الكلمة المفتاحية Dim

الكلمة المفتاحية Dim تخبر فيجوال بيسيك بأنك تريد أنشاء متغير , و المثال التالي يوضح كيفية استخدام الكلمة المفتاحية Dim عند إنشاء المتغير .

- Dim num_people As Integer
- Private num_people As Integer

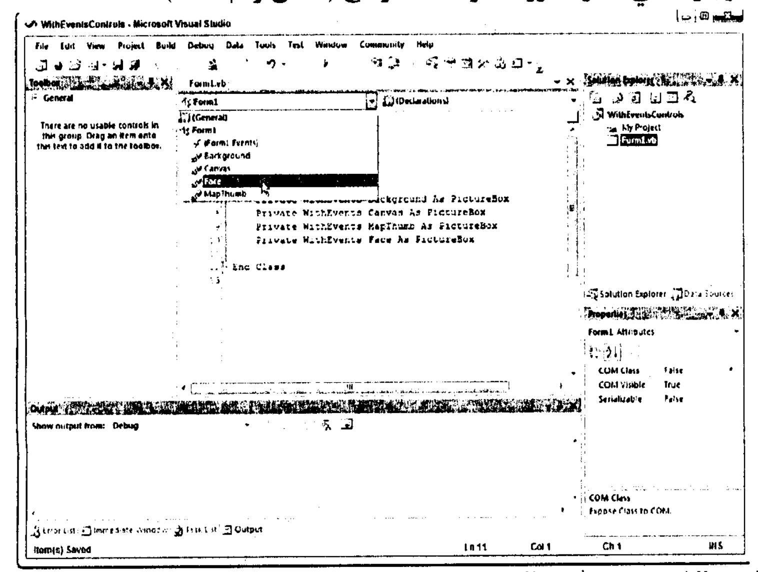
الكلمة المفتاحية WithEvents

تخبر هذه الكلمة المفتاحية أن المتغير الذي يتم إنشاؤه خاص بكائن

برمجي معين, و المثال التالي يوضح كيفية استخدام هذه الكلمة المفتاحية

- Private WithEvents Face As PictureBox

و عندما باستخدام الكلمة المفتاحية WithEvents عند تعريف المتغير تقوم فيجوال بيسيك بإنشاء قيد لهذه الكلمة المفتاحية في القائمة المنسدلة الموجودة في يسار محرر الأكواد كما يوضح (شكل رقم ٥-١) .



القيد الذي تقوم فيجوال بيسيك بإنشائه عند استخدام الكلمة المفتاحية WithEvents

(شكل رقم ٥-١)

تسمية المتغير

يوجد في فيجوال بيسيك قواعد معينه يجب أتباعها عند تسمية المتغير عند تعريفه في أول مرة , و هذه القواعد هي أنه يشترط أن يبدأ اسم المتغير

بحرف أو شرطة سفلية و لا يجوز أن يبدأ اسم المتغير برقم أو بأحد العلامات الخاصة , كما يمكن أن يحتوي أسم المتغير في داخله علي أرقام إلي جانب الأحرف التي يتكون منها الاسم , و في حالة ما إذا بدأ اسم المتغير بشرطة سفلية يجب أن يتبع هذه الشرطة حرف واحد علي الأقل , حيث لا يجوز أن يكون أسم المتغير عبارة عن شرطة سفلية فقط , و يمكن أن يكون أسم المتغير عبارة عن حرف واحد , و لكن لا يجوز أن يكون أسم المتغير عبارة عن رقم واحد أو مجموعة من الأرقام حيث لا يجوز أصلاً أن يبدأ برقم , و لا يجوز أن يحتوي اسم المتغير علي مسافات أو علامات خاصة , أو أن يكون أسم المتغير أحد الكلمات المحجوزة في اللغة .

و الجدول التالي يوضح بعض أمثلة لأسماء المتغيرات الصالحة و الغير صالحة و أساب عدم صلاحية الأسماء الغير صالحة .

أسم المتنا ر	الحالة	س ب عدم الصلاحية
Num_employees	صالح	
Num employees	غير صالح	يحتوي علي مسافة
NumEmployees	صالح	
Num*employee	غير صالح	يحتوي علي أحد العلامات الخاصة
_manger	صالح	
_	غير صالح	لتكونه من شرطة سفلية فقط
__	غير صالح	لعدم احتوائه علي حرف بعد الشرطة



أسم المتغ ر	الحالة	س ب عدم الصلاحية
		السفلية
2nd_employee	غير صالح	يبدأ برقم

جدول يوضح أمثلة علي أسماء المتغيرات

و كما سبق و أن ذكرنا لا يجوز أن يكون أسم المتغير عبارة عن أحد الكلمات المحجوزة أو الكلمات المفتاحية في فيجوال بيسيك , و لكن يمكنك أعطاء اسم المتغير أحد الكلمات المحجوزة أو أحد الكلمات المفتاحية في فيجوال بيسيك بوضع اسم المتغير بين قوسين [ ] و في هذه الحالة تتعامل فيجوال بيسيك معه علي أنه أسم متغير و ليس أحد الكلمات المفتاحية أو الكلمات المحجوزة كما في المثال التالي :

```
• Public Sub ParseString(ByVal [String] As String)
• Dim values() As String = Split([String])
• End Sub
```

المصفوفات

المصفوفة Array عبارة عن مجموعة من المتغيرات التي يتم تعريفها معاً , و المثال التالي يوضح كيفية تعريف مصفوفة مكونة من خمسة متغيرات عددية صحيحة .

```
• Dim a(4) As Integer
```

و العدد 4 الموجود في الكود السابق يمثل ترتيب العنصر الأخير في المصفوفة و التي تبدأ من 0 و 1 و 2 و هكذا حتى ترتيب العنصر الأخير في المصفوفة .

و في الكود السابق تم تعريف أسم المتغير الخامس في المصفوفة باسم a و كلمة Intger تحدد نوع القيمة التي يتم إعطاؤها لهذا المتغير بأنها عدد صحيح

و يمكنك الوصول إلي عناصر المصفوفة باستخدام ترتيب العنصر في المصفوفة و باستخدام أسم المصفوفة كما في الكود التالي :

- For i = 0 To 4
- Console.WritLine(a(i))
- Next

و هناك نوع من المصفوفات تسمي المصفوفات متعددة الأبعاد Multi Dimensional Array و التي تحتوي علي مصفوفات أخري كعناصر لهذه المصفوفة بدلاً من احتوائها علي متغيرات , و تتعدد درجات المصفوفات متعددة الأبعاد , فهناك المصفوفات ثنائية الأبعاد و التي تحتوي علي مصفوفات كعناصر للمصفوفة , و مصفوفات ثلاثية الأبعاد و التي تحتوي علي مصفوفات ثنائية الأبعاد كعناصر للمصفوفة , و الكود التالي يقوم بتعريف مصفوفة ثلاثية الأبعاد , تحتوي علي عناصر عبارة عن مصفوفات ثنائية الأبعاد بأحجام مختلفة .

- Dim a(5, 10, 15) As Integer

تحديد عناصر المصفوفة

يتم تحديد عناصر المصفوفة عن طريق وضع عناصر المصفوفة بين قوسين { } كما في الكود التالي :

- Dim a( ) As String = {"First", "Second", "Third", "Fourth", "Fifth"}

و في الكود السابق لم يتم تعريف عناصر المصفوفة عن طريق عدد هذه العناصر و لكن تم تحديد هذه العناصر بأسمائها و وضع هذه الأسماء بين

القوسين {} و في هذه الحالة يقوم مفسر الأكواد باستنتاج عدد عناصر المصفوفة عن طريق عدد العناصر المحدد أسماؤها في المصفوفة , و لتحديد عناصر المصفوفة متعددة الأبعاد بأسماء العناصر التي تحتوي عليها المصفوفة يتم وضع عناصر كل مصفوفة من المصفوفات ذات البعد الواحد التي تعتبر عناصر في المصفوفة متعددة الأبعاد بداخل الأقواس { } و يتم وضع كل هذه المصفوفات في داخل الأقواس { } كما في الكود التالي :

```
• Dim a(,) As Integer = {{1, 2}, {3, 4}, {5, 6}}
```

و ينتج عن تعريف المصفوفة متعددة الأبعاد في الكود السابق المصفوفة التالية :

```
• a(0,0)=1 a(0,1)=2
• a(1,0)=3 a(1,1)=4
• a(2,0)=5 a(2,1)=6
```

الكلمة المفتاحية New

تستخدم الكلمة المفتاحية New في توزيع عناصر أي نوع من المصفوفات , فعلي سبيل المثال الكود التالي يقوم بإنشاء مصفوفة من خمسة متغيرات عددية صحيحة و يقوم بتوزيع عناصر هذه المصفوفة

```
• Dim a( ) As Integer
• a = New Integer(4) {1, 2, 3, 4, 5}
```

كما يمكن أيضاً استخدام الأقواس { } في تحديد عناصر المصفوفة التي تم توزيع عناصرها بواسطة الكلمة المفتاحية New كما يلي :

```
• Dim a( ) As Integer
• ' allocates an uninitialized array of five Integers
    • a = New Integer(5) {}
```



الاسم الكامل

يستخدم الاسم الكامل Namespaces عند كتابة الأكواد في فيجوال بيسيك لمنع التعارض بين الأسماء المتشابهة للمتغيرات و الدوال و الفئات , ففي حالة وجود أكثر من متغير يحمل نفس الاسم ولكن كل متغير من هذه المتغيرات موجود في فئة معينة ففي هذه الحالة عند قيام مفسر الأكواد في تفسير الأكواد التي تحتوي علي هذه المتغيرات التي تحمل نفس الاسم فإن مفسر الأكواد لا يستطيع أن يعرف أي المتغيرات يتم استخدامه , و لجعل مفسر الأكواد يستطيع التعامل مع الأسكريبت يلزم استخدام الاسم الكامل للمتغير و الاسم الكامل للمتغير هو عبارة عن أسم المتغير متبوع باسم الفئة الموجود فيها هذا المتغير .

و عي سبيل المثال أفترض أنه قد تم تعريف أحدي الفئات و التي تسمي SomeClass باستخدام الكود التالي :

- Public Class SomeClass
- End Class

و الفئة SomeClass التي تم تعريفها في الكود السابق يمكن أن تكون موجودة في مكتبة الفئات الموجودة في البرنامج أو في بعض الملفات التي قام المستخدم بكتابتها , علي ذلك تكون نفس الفئة SomeClass موجودة أكثر من مرة في البرنامج و في مكانين مختلفين أو أكثر , و في هذه الحالة لن يستطيع مفسر الأكواد أن يعرف أي الفئات SomeClass يختار للتعامل معها لوجود هذه الفئة أكثر من مرة في البرنامج و في مواضع مختلفة , و في هذه الحالة يجب علي المبرمج أن يقوم بكتابة أسم الفئة كاملاً و الذي يعرف باسم

namespace عند استخدام هذه الفئة في البرنامج كما في الكود التالي :

```
' The namespace is "FooBarCorp.SuperFoo2100".
Dim x As New FooBarCorp.SuperFoo2100.SomeClass( )
x.DoSomething( )
' The namespace is "MegaBiz.ProductivityTools.WizardMaster".
Dim y As New MegaBiz.ProductivityTools.WizardMaster.SomeClass( )
y.DoSomethingElse( )
```

المعاملات Operators

المعاملات عبارة عن رموز أو كلمات مفتاحية أو علامات خاصة أو أحرف تستخدم في الكود لتحديد العملية التي يقوم الكود بأدائها , و من أمثلة هذه المعاملات علامة + التي تستخدم في جمع القيم و علامة * التي تستخدم في ضرب القيم بعضها في بعض , و هناك نوعين من المعاملات , هناك المعاملات الفردية Unary Oprators و تكون عبارة عن معامل واحد , و هناك معاملات زوجية binary و تكون عبارة عن معاملين معاً .

المعاملات الفردية

تدعم فيجوال بيسيك المعاملات الفردية مثل علامة + و تستخدم هذه العلامة في الجمع , و علامة - و تستخدم في الطرح و التعبير عن القيم السالبة كما و المعامل AddressOf و الكود التالي يوضح كيفية استخدام المعامل المفرد .

```
Dim sh As Short = -32768
Dim i As Integer = -sh
```



المعاملات الحسابية

تستخدم المعاملات الحسابية في أداء العمليات الحسابية المعروفة و التي يتم إجراؤها علي الأرقام مثل عملية الضرب و عملية القسمة و الجمع و الطرح .

فعلامة * تستخدم في ضرب الأعداد بعضها في بعض , و علامة / تستخدم في القسمة , و علامة \ تستخدم في قسمة الأعداد الصحيحة , و علامة ^ تستخدم للرفع إلي أس , و علامة + و علامة - تستخدم في الجمع و الطرح .

معاملات المقارنة

تستخدم معاملات المقارنة Relational Opreators في المقارنة بين القيم بعضها ببعض , و تعود هذه المعاملات بقيمة منطقية Boolean Value أما صح True أو خطأ False

و من علامات المقارنة علامة التساوي = التي تستخدم لفحص تساوي القيم مع بعضها البعض و علامة التباين < > و التي تستخدم للتحقق من عدم تساوي القيم بعضها مع بعض , وعلامة أقل من > و أكبر من < و التي تستخدم للمقارنة بين القيم الغير متساوية , و علامة أقل من أو يساوي => و علامة اكبر من أو يساوي =< و التي تستخدم في التعرف علي القيم المتساوية أو المتباينة .

المعامل TypeOf...Is

يستخدم المعامل TypeOf...Is في التعرف علي أنواع المتغيرات , و يعود



هذا المعامل بالنتيجة True في حالة ما إذا كان المتغير الذي نقوم بفحص نوعه من نفس النوع الذي نقوم بفحصه , و يعود بالنتيجة False إذا كان من نوع مختلف .

المعامل Is

يستخدم هذا المعامل أيضاً للاستفسار عن أنواع المتغيرات , فلو كانت الإجابة عن الاستفسار بالإيجاب يعود بالنتيجة True و لو كانت الإجابة بالنفي يعود بالنتيجة False

المعامل Like

يستخدم المعاملLike في فحص المتغيرات اللفظية , و مدي مطابقتها مع لفظ معين , فإذا كان المتغير اللفظي مطابق للفظ الذي يتم المطابقة به يعود هاذ المعامل بالنتيجة True و إن لم يكن مطابقاً له يعود بالنتيجة False

المعامل And

يستخدم في المقارنات المنطقية و يعود بالنتيجة True إذا كانت كل المتغيرات التي يتم مقارنتها يتحقق فيها شرط معين .

المعامل AndAlso

يستخدم في المقارنة المركبة , و يعود بالنتيجة True في حالة ما إذا كانت كل نتائج المقارنة صحيحة , و يعود بالنتيجة False إذا كانت أحد نتائج المقارنة خاطئة .



المعامل Or

يستخدم في المقارنات المنطقية و يعود بالنتيجة True إذا كان أحد المتغيرات علي الأقل يحقق الشرط المطلوب .

المعامل OrElse

يستخدم في المقارنات المنطقية و يعود بالنتيجة True إذا كان أحد المتغيرات علي الأقل يحقق الشرط المطلوب .

الجمل البرمجية Statements

لغة فيجوال بيسيك لغة سطرية Line Oriented Language بمعني أن كل سطر برمجي يمثل أمراً مستقلاً بزاته , و علي ذلك فالجمل البرمجية هي الوسيلة التي تجعل مجموعة من الأسطر التي تتكون منها الجملة بلوك برمجي مستقل بزاته و ليس كل سطر بداخله مستقل عن الأخر , فالجملة Statement عبارة عن عدد من الأسطر البرمجية المتتالية تعمل كلها معاً بشكل متكامل كجزء واحد .

و تستخدم الجملة البرمجية الشرطة السفلية لتخبر مفسر الأكواد أن مجموعة الأسطر التي تتوسطها هذه الشرطة عبارة عن جملة واحدة يتم تنفيذها معاً كما في الجملة التالية :

```
● Dim strSql As String = "SELECT Customers.CompanyName," _
● & " COUNT(Orders.OrderID) AS OrderCount" _
● & " FROM Customers INNER JOIN Orders" _
● & " ON Customers.CustomerID = Orders.CustomerID" _
● & " GROUP BY Customers.CompanyName" _
● & " ORDER BY OrderCount DESC"
```

كما يمكن استخدام النقطتين Clone : في وضع أكثر من جملة برمجية



في سطر واحد كما يلي :

● i = 5 : j = 10

جمل الاختيار

في فيجوال بيسيك يوجد ثلاثة جمل اختيار Option Statement تؤثر في سلوك مفسر الأكواد , و في حالة استخدام أي من هذه الجمل الثلاثة يجب أن تظهر الملف الذي يحتوي علي الأكواد المستخدم فيه هذه الجملة , و تتحكم جملة الاختيار في قيام مفسر الأكواد بتفسير الكود الموجود في الملف الموجود فيه هذه الجملة .

و جمل الاختيار الثلاثة الموجودة في فيجوال بيسيك هي جملة Option Comapare و جملة Option Explicit و جملة Option Strict

و تستخدم جملة Option Compare في تحديد الطريقة التي يتم بها مقارنة المتغيرات النصية مع بعضها البعض كما يلي :

● Option Compare [Binary | Text]

و الكود السابق يوضح أن جملة الاختيار Option Compare لها حالتان أو خيارات هما الخيار Binary و الخيار Text و يستخدم أي من هذين الخيارين في ملف الأكواد و ليس كلاهما , ففي حالة استخدام الاختيار Binary فهذا الخيار معناه أننا نقول لمفسر الأكواد أن يفرق بين الأحرف الكبيرة و الأحرف الصغيرة عند مقارنة القيم اللفظية مع بعضها البعض , أما عند استخدام الاختيار Text ففي هذه الحالة لا يقوم مفسر الأكواد بالنظر إلي الأحرف الكبيرة و الأحرف الصغيرة بل يقوم بمقارنة النصوص علي أساس الأحرف كبيرة أو

صغيرة .

و جملة Option Explicit تستخدم لتحدد ما إذا كان مفسر الأكواد يحتاج إلي تعريف كل متغير من المتغيرات المستخدمة في الأكواد صراحة أم لا , و تستخدم هذه الجملة كما يلي :

• Option Explicit [On | Off]

و كما يتضح من الكود السابق أن الجملة تحتوي علي خيارين هما الخيار On و الخيار Off و في حالة استخدام الخيار On ففي هذه الحالة يتطلب مفسر الأكواد تعريف كل المتغيرات أولاً قبل استخدامها حتى يستطيع التعرف عليها , أما في حالة استخدام الخيار Off يعتبر مفسر الأكواد أن الكود الخاص باستخدام المتغير هو كود تعريف المتغير , و أن القيمة التي تعطي للمتغير في هذا الكود هي التي تحدد نوع هذا المتغير .

أما جملة Option Strict فهذه الجملة هي التي تحدد ما إذا كان مفسر الأكواد يحتاج أن نقوم بتحديد نوع المتغير صراحة عند تعريف المتغير أم لها و تستخدم هذه الجملة كما يلي :

• Option Strict [On | Off]

و كما يتضح من الكود السابق أن جملة Option Strict لها خيارين هما On و Off و باستخدام خيار On في هذه الحالة يحتاج مفسر الأكواد أن نقوم صراحة بتحديد نوع المتغير صراحة عند تعريف المتغير و إلا فلن يستطيع مفسر الأكواد التعرف علي نوع المتغير , أما باختيار الخيار Off ففي هذه الحالة يمكن عدم تحديد نوع المتغير صراحة عند تعريف المتغير و يستطيع مفسر

الأكواد تخمين نوع المتغير من نوع القيمة المعطاة لهذا المتغير .

جمل التكرار

تدعم فيجوال بيسيك عدد كبير من جمل التكرار Branching Statements التي تقوم بأداء سلسلة من العمليات المتتالية , و من جمل التكرار التي تدعمها فيجوال بيسيك ما يلي :

جملة Call

تستخدم جملة Call في استدعاء الدوال البرمجية الخارجية عن الأسكريبت و تنفيذها في الأسكريبت , و تستخدم هذه الجملة كما يلي :

- SomeMethod( )

جملة Exit

تستخدم جملة Exit في إيقاف تنفيذ بلوك معين من الأكواد الموجودة فيه هذه الجملة و الخروج من هذا البلوك , و تستخدم هذه الجملة عادة في كسر حلقات التكرار بتحقق شرط معين , و تأخذ جملة Exit عدة أشكال معينة كما يلي :

جملة Exit Do

تقوم هذه الجملة بإيقاف التكرار الناتج من استخدام جملة التكرار Do

جملة Exit For

تقوم هذه الجملة بإيقاف التكرار الناتج من استخدام جملة For

جملة Exit Function

تستخدم هذه الجملة في إيقاف تنفيذ الدوال .



جملة Exit Property

تقوم هذه الجملة بإيقاف تنفيذ الأكواد الخاصة بتحديد الخواص و الجمل الخاصة بفحص الخواص .

جملة Goto

تقوم جملة Goto بتحويل عملية تنفيذ الأكواد إلي البلوك التابع لهذه الجملة كما يلي :

```
• Goto MyLabel
• MyLabel:
```

جملة If

تستخدم جملة If في وضع تنفيذ مجموعة من الأكواد قيد تحقيق شرط معين , فبتحقق هذا الشريط يتم تنفيذ مجموعة الأكواد , و بانعدام تحقيق هذا الشريط لا يتم تنفيذ الأكواد و يتم الخروج من هذه المجموعة من الأكواد , و الصيغة العامة لاستخدام جملة If هي كما يلي :

```
• If expression Then
• statements
• End If
```

و في الصيغة السابقة حيث expression عبارة عن الشرط الذي يتم وضعه لجملة If و يتم وضع هذا الشريط في صيغة منطقية تأتي بالقيمة True و في هذه الحالة يتم تنفيذ مجموعة الأكواد المقصود بها Statement في الصيغة , أو يأتي الشرط المنطقي في بالقيمة False في حالة عدم تحققه و في هذه الحالة يتم الخروج بدون تنفيذ مجموعة الأكواد الموجودة في الصيغة .

و يمكن استخدام مجموعة أخري من الأكواد يتم تنفيذها مع جملة If في حالة عدم تحقق الشرط المرتبط بجملة If و ذلك باستخدام جملة Else كما يلي :

- If expression Then
- statements
- Else
- statements
- End If

ففي الصيغة السابقة إذا كان الشرط Expression صحيح و تحقق هذا الشرط يتم تنفيذ مجموعة الأكواد Statements و إن لم يتحقق هذا الشرط يتم الخروج من هذه المجموعة من الأكواد بدون تنفيذها , و يتم تنفيذ مجموعة الأكواد Statements الخاصة بجملة Else

كما يمكن أيضاً وضع أكثر من شرط في جملة If باستخدام جملة ElseIf كما في الصيغة التالية :

- If expression Then
- statements
- ElseIf expression Then
- statements
- ElseIf expression Then
- statements
- Else
- statements
- End If

ففي الصيغة السابقة إذا تحقق الشرط الخاص بجملة If يتم تنفيذ مجموعة الأكواد التالية لجملة If و أن لم يتحقق هذا الشرط يتم البحث في الشرط الأخر الموجود في جملة ElseIf فإن تحقق يتم تنفيذه , و أن لم يتحقق يتم

تنفيذ الشرط الموجود في جملة Else

جملة RaisEvent

تقوم جملة RaiseEvent بتشغيل حدث معين و تنفيذ الأوامر التالية لهذا الحدث مباشرة .

جملة Return

تستخدم جملة Return في الخروج من تنفيذ دالة معينة بناتج تنفيذ هذا الدالة في شكل قيمة يمكن استخدامها مرة أخري في الأكواد , و الكود التالي يوضح كيفية استخدام جملة Return في الخروج بناتج تشغيل دالة MyFactorial في شكل قيمة عددية يتم وضعها كقيمة للمتغير Retval الذي يتم تعريفه في السطر الثاني من الكود .

- Public Shared Function MyFactorial(ByVal value As Integer) As Integer
- Dim retval As Integer = 1
- Dim i As Integer
- For i = 2 To value
- retval *= i
- Next
- Return retval
- End Function

و الكود التالي مثال أخر لاستخدام جملة Return في الخروج بناتج تشغيل دالة MyFractorial في شكل قيمة

- Public Shared Function MyFactorial(ByVal value As Integer) As Integer
- Dim retval As Integer = 1
- Dim i As Integer
- For i = 2 To value
- retval *= i
- Next



- MyFactorial = retval
- End Function

جملة Select Case

تستخدم جملة Select Case في تنفيذ أمر معين عند تحقق قيمة معينه لمتغير , و في الصيغة التالية الخاصة باستخدام جملة Select Case يتم قيمة المتغير اللفظي StrColor الموجود في السطر الأول من الكود , ففي حالة ما إذا كانت قيمة هذا المتغير هي Red كما في السطر الثاني يتم تنفيذ الأمر المشار إليه بمجموعة النقاط الموجودة في السطر الثالث , و في حالة ما إذا كانت قيمة هذا المتغير green كما في السطر الرابع يتم في هذه الحالة تنفيذ الأمر الموجود في السطر الخامس و المشار إليها بمجموعة النقاط و هكذا .

- Select Case strColor
- Case "red"
- Case "green"
- Case "blue"
- Case "yellow"
- Case Else
- End Select

جملة Do

تقوم الجملة التكرارية Do بتنفيذ أمر معين بشكل متكرر طالما تحقق شرط معين , و يتم تنفيذ هذه الأكواد بشكل مرات متتالية و الخروج بقيمة معينه , و مقارنة هذه القيمة مع القيمة المحددة في الجملة , فإذا كانت نتيجة المقارنة True يتم تنفيذ الأمر مرة أخري و مقارنة القيمة مرة أخري , و هكذا حتى ينتفي تنفيذ هذا الشرط , و فيما يلي مجموعة من الطرق

لاستخدام جملة Do

```
Do While i < 10
Loop
Do Until i >= 10
Loop
Do
Loop While i < 10
Do
Loop Until i >= 10
Do
Loop
```

و الصيغة السابقة تمثل أكثر من طريقة لاستخدام جملة Do التكرارية , ففي الصيغة الأولي و هي عبارة عن الثلاثة أسطر الأولي يتم تنفيذ الأوامر الممثلة بالنقاط طالما تحقق الشرط و كانت قيمة المتغير i أقل من 10 و باقي الصيغ الممثلة في باقي الأكواد تحقق نفس النتيجة و لكن بصيغ مختلفة لاستخدام جملة Do

جملة For

تستخدم جملة التكرار For في تنفيذ مجموعة معينه من الأوامر لعدد معين من المرات يتم تحديد هذا العدد من المرات علي أساس قيمة متغير معين , و فيما يلي الصيغة الخاصة باستخدام جملة For

```
For variable = expression To expression [ Step expression ]
statements
Next [ variable_list ]
```

في الصيغة السابقة في السطر الأول يتم وضع أسم المتغير Variable بعد كلمة For و يتم تحديد قيمة هذه المتغير التي تعتبر هي عدد المرات التي يتم

فيها تنفيذ مجموعة الأوامر بين القوسين [ ] و يتم وضع الأوامر التي يتم تنفيذها في السطر الثاني و المشار إليها بكلمة statements

و الكود التالي مثال للطريقة التي تستخدم بها جملة For

- For i = 1 To 10
- For j = 1 To 10
- For k = 1 To 10
- Next k, j, I

في الكود السابق في السطر الأول تم أستخدم جملة For لثلاثة متغيرات هي المتغير i و المتغير j و المتغير k و تم تحديد عدد مرات تنفيذ الأوامر بعشرة مرات , و تم الإشارة إلي هذه الأوامر بالنقاط الموجودة في السطر الرابع من الكود .

و يمكن كتابة الكود السابق علي النحو التالي أيضاً

- For i = 1 To 10
- For j = 1 To 10
- For k = 1 To 10
- Next
- Next
- Next

جملة For Each

تتشابه جملة التكرار For Each مع جملة For و لكن في جملة For Eeach لا تحتاج إلي تحديد عدد مرات تنفيذ الأكواد في شكل عددي يعطي كقيمة للمتغير كما في حملة For و فيما يلي الصيغة الخاصة باستخدام جملة For Each

- For Each variable In expression
- statements

```
• Next [ variable ]
```

و معني الكود السابق هو تنفيذ الأوامر الموجودة في السطر الثاني عدد من المرات يساوي عدد المتغيرات الموجودة في الكود .

و فيما يلي مثال علي كيفية استخدام جملة For Each

```
• Dim a( ) As Integer = {1, 2, 3, 4, 5}
• Dim b As Integer
• Dim i As Integer
• For i = a.GetLowerBound(0) To a.GetUpperBound(0)
• b = a(i)
• Console.WriteLine(b)
• Next
```

و يمكن كتابة المثال السابق بالطريقة التالية :

```
• Dim a( ) As Integer = {1, 2, 3, 4, 5}
• Dim b As Integer
• For Each b In a
• Console.WriteLine(b)
• Next
```

الفئات Classes

الفئات هي أحد أنواع البيانات , و تستخدم الفئات في البرمجة كائنية التوجيه للتعامل مع مجموعة معينة من البيانات المتعلقة بموضوع واحد كمجموعة البيانات المتعلقة بأحد العملاء أو أحد العاملين بأحدي الشركات , و يتم وضع البيانات الخاصة بالعامل أو بالعميل في شكل قيم متغيرات كالاسم و السن و الوظيفة و الراتب , كل هذه البيانات الخاصة بالعامل مثلاُ يتم تعريفها كمتغيرات و وضعها ضمن الفئة Class التي تضم بيانات هذا العامل , و الفئة لا تعتبر كائن برمجي في حد ذاتها و أنما العميل أو العامل أو الطلبية هي التي تعتبر كائن , و



تتضمن عملية تعريف الفئة عملية تعريف جماعية لمجموعة من المتغيرات معاً هذه المتغيرات التي تضمها الفئة تمثل خواص أو بيانات الكائن البرمجي و هو العامل أو الطلبية أو العميل .

و المثال التالي يمثل أحد الأكواد المستخدمة في تعريف فئة تسمي Employee تضم كل البيانات المتعلقة بأحد العاملين .

```
• Public Class Employee
• Public EmployeeNumber As Integer
• Public FamilyName As String
• Public GivenName As String
• Public DateOfBirth As Date
• Public Salary As Decimal
• Public Function Format( ) As String
• Return GivenName & " " & FamilyName
• End Function
• End Class
```

في السطر الأول من الكود السابق تم تعريف الفئة Employee و في السطر الثاني من الكود تم تعريف المتغير EmployeeNumber الذي يمثل رقم العامل كمتغير عددي صحيح , و في السطر الثالث تم تعريف المتغير الخاص باسم عائلة العامل أو لقب العامل FamilyName كمتغير لفظي , و هكذا في السطر الرابع و الخامس و السادس تم تعريف باقي بيانات العامل في شكل متغيرات .

و الكود التالي يوضح مثالاً علي كيفية استخدام الفئات في تخزين البيانات الخاصة بأحد العمال و التعامل مع هذه البيانات .

```
• Dim emp As New Employee( )
• emp.EmployeeNumber = 10
```



- emp.FamilyName = "Rodriguez"
- emp.GivenName = "Celia"
- emp.DateOfBirth = #1/28/1965#
- emp.Salary = 115000
- Console.WriteLine("Employee Name: " & emp.Format( ))
- Console.WriteLine("Employee Number: " & emp.EmployeeNumber)
- Console.WriteLine("Date of Birth: " & emp.DateOfBirth.ToString("D", Nothing))
- Console.WriteLine("Salary: " & emp.Salary.ToString("C", Nothing))

و تكون نتيجة تشغيل الكود السابق هو الخروج ببيانات العامل كما يلي :

- Employee Name: Celia Rodriguez
- Employee Number: 10
- Date of Birth: Thursday, January 28, 1965
- Salary: $115,000.00

الحقول

تحتوي الحقول Fields علي البيانات الأساسية الخاصة بالكائن البرمجي و يتم إنشاء الحقول ضمن الكود الخاص بتعريف الفئة , و القيم التي يتم تخزينها في هذه الحقول هي القيم المتعلقة بالكائن البرمجي الذي تمثله الفئة كما يلي :

- Dim emp1 As New Employee( )
- Dim emp2 As New Employee( )
- emp1.EmployeeNumber = 10
- emp2.EmployeeNumber = 20 ' Doesn't affect emp1.

كما يمكن استخدام الكلمة المفتاحية Shared عند تعريف الحقول حتى تستطيع استخدام نفس الحقول الواحدة في أكثر من فئة كما في الكود التالي

- Public Class X
- Public Shared a As Integer
- End Class



و في هذه الحالة بتغيير أحد قيم الحقول في أحدي الفئات يتم تغيير قيمة الحقل في كل الفئات المشاركة في هذا الحقل .

الحقول للقراءة فقط

يمكنك أن تقوم بجعل الحقول حقول للقراءة فقط Read-only Fields عند تعريف هذه الحقول في الكود الخاص بإنشاء الفئة التي تحتوي علي هذه الحقل و ذلك باستخدام عبارة ReadOnly كما في الكود التالي :

```
• Public Class MyDataTier
• Public ReadOnly ActiveConnection As System .Data .SqlClient SqlConnection
• Public Sub New(ByVal ConnectionString As String)
• ActiveConnection = _
• New System.Data.SqlClient.SqlConnection(ConnectionString)
• End Sub
• End Class
```

الأحداث

يمكن استخدام أحد حقول الفئات للتعبير عن حدث Event كما في الكود التالي :

```
• Imports System.Data.SqlClient
• Public Class EventHandlingTest
• Private WithEvents m_cn As SqlConnection
• Public Sub MySqlInfoMessageEventHandler( _
• ByVal sender As Object, _
• ByVal e As SqlInfoMessageEventArgs _
• ) Handles m_cn.InfoMessage
• Dim sqle As SqlError
• For Each sqle In e.Errors
• Debug.WriteLine(sqle.Message)
```



- Next
- End Sub
- End Class

في الكود السابق تم تعريف الفئة EventHandlingTest في السطر الثاني من الكود , و في السطر الثالث من الكود تم تعريف الحدث m_cn ليستخدم للاتصال بقواعد البيانات SqlConnection و في هذا السطر تم تعريف الحدث باستخدام الكلمة المفتاحية WithEvents حتى تستطيع الفئة أن تقوم باستقبال الاتصال بقواعد البيانات .





طرق جمع البيانات من المستخدم

مربعات الاختيار CheckBoxes
قوائم الاختيار Menues
مربعات الحوار Dialog Boxes

مربعات الاختيار CheckBoxes

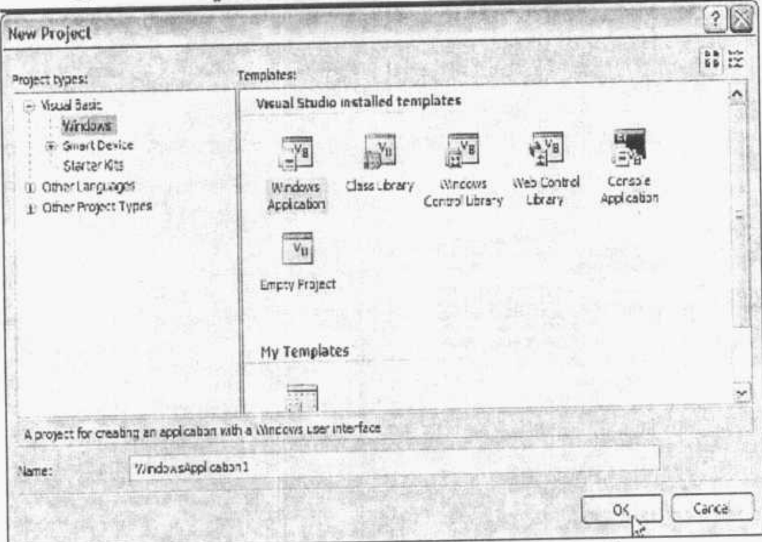
تحتوي فيجوال بيسيك علي طرق عديدة لجمع البيانات من المستخدم و التعامل مع هذه البيانات , فالحقول النصية التي تحتوي عليها النماذج من أحد الطرق التي يمكن استخدامها في جمع البيانات من المستخدم عند طريق قيام المستخدم بالكتابة مباشرة في هذه الحقول النصية , و قوائم الاختيار أيضاً من وسائل جمع البيانات من المستخدم عن طريق قيام المستخدم بالنقر علي أحد العناصر التي تحتوي عليها القائمة , كما تحتوي مربعات الحوار أيضاً علي العديد من وسائل جمع البيانات من المستخدم , و في هذا الفصل سوف نعرض لأهم وسائل لجمع البيانات من المستخدم مثل أزرار الاختيار RadioButtons و مربعات الاختيار CheckBoxes و قوائم الاختيار ListBox و ذلك من خلال شرح خطوات مشروع يضم هذه الوسائل الخاصة بجمع البيانات من المستخدم.

و لتنفيذ هذا المشروع , قم بإغلاق أي مشروع مفتوح في برنامج فيجوال أستوديو , و قم بإنشاء مشروع جديد باختيار New من قائمة File و منها اختار Project ليظهر بذلك المربع الحواري الخاص بإنشاء مشروع جديد كما هو موضح في (شكل رقم ٧-١) .

في المربع الحواري الخاص بإنشاء مشروع جديد الموضح في الشكل قم بالنقر علي أيقونة Windows Application و أكتب أسم المشروع الجديد MyCheckBox في حقل Name و اختار المسار الذي تريده لحفظ ملفات المشروع , و أضغط زر Ok ليقوم فيجوال أستوديو بإنشاء الملفات الخاصة بإنشاء



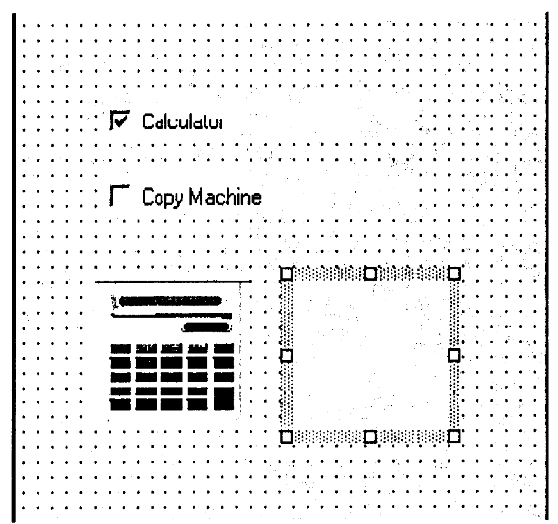
المشروع الجديد في هذا المسار , و يظهر نموذج فارغ في نافذة المشروعات .



المربع الحواري الخاص بإنشاء مشروع جديد
(شكل رقم ٧-١)

و في شريط أدوات عناصر التحكم في النموذج الموجود في يسار واجهة البرنامج قم بالنقر فوق ChecBox و قم بالسحب بزر الماوس فوق النموذج لإدراج مربعين اختيار في النموذج كما يوضحهما (شكل رقم ٧-٢) , ثم قم بالنقر فوق PictureBox في شريط أدوات عناصر التحكم , و قم بالسحب فوق النموذج لإدراج مربعين صورة كالموضحين في الشكل .





النموذج المستخدم في عمل المشروع

(شكل رقم ٧-٢)

و في نافذة الخواص قم بضبط الخيارات الموضحة في الجدول التالي و الخاصة بعناصر التحكم الأربعة التي قمت بإضافتها للمشروع .

العنصر	الخاص :	القيمة
ChecBox1	Checked Text	True "Calculator"
CheckBox2	Text	"Copy Machine"
PictureBox1	Image Size Mode	مسار الصورة
PictureBox2	Size Mode	StretchImage

جدول يوضح قيم خواص عناصر النموذج

و في هذا المشروع سوف نستخدم مربعات الاختيار في إظهار و إخفاء صور الآلة الحاسة و الطابعة , و تحدد قيمة خاصية Text الخاصة بمربع الاختيار



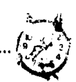
تحدد محتويات هذا المربع , و تتيح لك الخواص الافتراضية لمربع الاختيار أن تقوم بتحديد أو عدم تحديد مربع الاختيار باختيار True أو False من بين الخيارات الخاصة بهذه الخاصية , و في الجدول السابق الخاص بخواص أدوات التحكم قمنا بوضع قيمة خاصية Size Mode الخاصة بمربع الصورة عند القيمة StrechImage لكي تتمدد الصورة أو تنكمش بقدر حجم مربع الصورة .

تحديد خواص عناصر التحكم في النموذج

لتحديد خواص عناصر التحكم في النموذج التي قمت بإضافتها , أبدأ أولاً بمربع الاختيار الأول CheckBox1 الموجود في أعلي النموذج , و أنقر نقراً مزدوجاً فوق هذا المربع لفتح محرر الأكواد يحتوي علي الأكواد الخاصة بهذا المربع , و بين السطرين الموجودين في محرر الأكواد الخاصين بوظيفة هذا الزر قم بكتابة الكود التالي :

```
• If CheckBox1.CheckState = 1 Then
• PictureBox1.Image = System.Drawing.Image.FromFile _
• ("c:\vbnetsbs\chap03\calcultr.bmp")
• PictureBox1.Visible = True
• Else
• PictureBox1.Visible = False
• End If
```

و الكود السابق يجعل مربع الاختيار يستقبل البيانات من المستخدم عن طريق تحديد أو عدم تحديد هذا المربع , ففي حالة قيام المستخدم بتحديد هذا المربع تظهر صورة الآلة الحاسبة في مربع الصورة الموجود إلي اليسار و الموضوعة في المسار المحدد في الكود و بنفس الاسم .

و لتحديد خواص الزر الثاني CheckBox2 قم بإظهار نمط التصميم في واجهة البرنامج بالنقر علي زر إظهار نمط التصميم ليظهر النموذج مرة أخري , و في النموذج أنقر نقراً مزدوجاً فوق مربع الاختيار الثاني ليتم عرض محرر الأكواد و يعرض الأكواد الخاصة بوظيفة هذا الزر , و بين الأسطر الأول و الأخير الخاصين بوظيفة هذا الزر قم بكتابة الكود التالي :

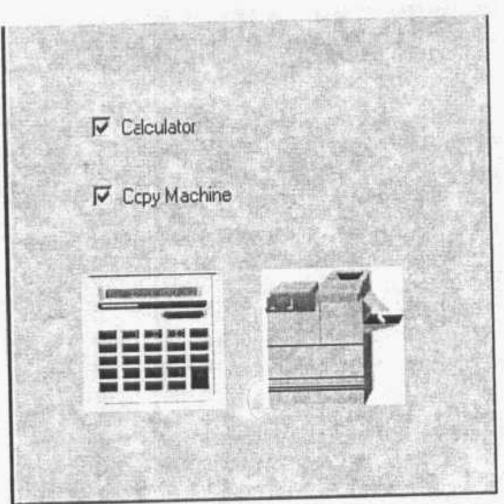
- If CheckBox2.CheckState = 1 Then
- PictureBox2.Image = System.Drawing.Image.FromFile _
- ("c:\vbnetsbs\chap03\copymach.bmp")
- PictureBox2.Visible = True
- Else
- PictureBox2.Visible = False
- End If

ثم قم بحفظ المشروع بالنقر فوق زر Save All في شريط الأدوات .

تجربة المشروع

لتجربة المشروع قم بالنقر فوق زر Sart في شريط الأدوات ليقوم فيجوال بيسيك بعرض المشروع علي النحو الذي تستطيع تجربته فيه , كما يوضح (شكل رقم ٧-٣) , و في النموذج قم بتحديد مربع الاختيار الأول لتظهر صورة الآلة الحاسبة و قم بتحديد مربع الاختيار الثاني لتظهر صورة الطابعة .





عرض المشروع للتجربة

(شكل رقم ٧-٣)

ثم قم بإغلاق المشروع بالضغط علي علامة X الموجودة أعلي يمين النموذج .

قوائم الاختيار Menues

قائمة الاختيار عبارة عن قائمة تظهر عناصرها بالنقر علي أسم هذه القائمة , و تحتوي قائمة الاختيار علي مجموعة من الأوامر , و هناك نوعين من الأوامر يمكن أن تحتوي عليهما قائمة الاختيار , النوع الأول هو الأوامر التي يتم تنفيذها بمجرد اختيارها من القائمة , و النوع الثاني من الأوامر التي تحتوي عليها قائمة الاختيار هي الأوامر التي باختيارها يتم فتح مربع حواري يحتوي

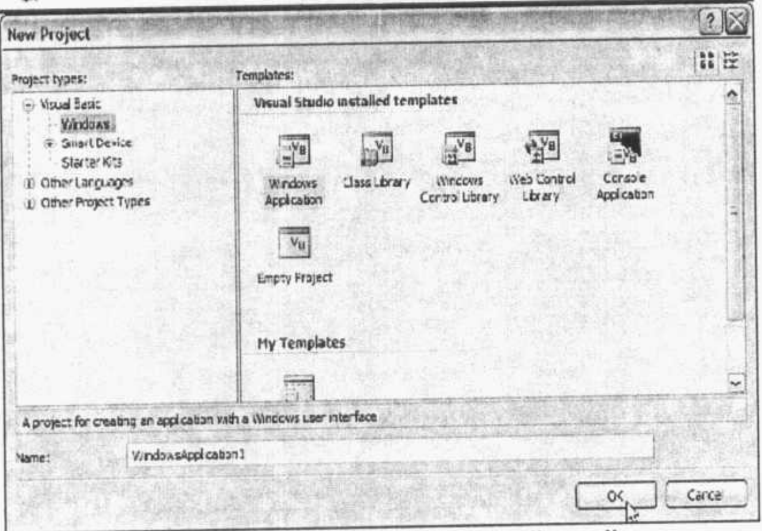
علي عدة خيارات متعلقة بتنفيذ الأمر الموجود في القائمة .

مشروع قوائم الاختيار

فيما يلي تطبيق بسيط علي كيفية أنشاء قوائم الاختيار , التطبيق عبارة عن نموذج يحتوي علي قائمة , هذه القائمة تستخدم في إنشاء ساعة , و باختيار الأوامر التي تحتوي عليها هذه القائمة يتم إظهار الوقت و التاريخ .

لبدء تنفيذ المشروع أختار New من قائمة File و منها اختار Project ليظهر المربع الحواري الخاص بإنشاء مشروع جديد كما هو موضح في (شكل رقم ٧-٤) .

في المربع الحواري الخاص بإنشاء مشروع جديد الموضح في الشكل , اختار Windows Application من نافذة أنواع المشروعات الموجودة في يمين المربع الحواري , و قم بكتابة أسم المشروع الجديد MyMenu في حقل Name في المربع الحواري و اختار المسار الذي تريد أن يقوم فيجوال أستوديو بحفظ الملفات الخاصة بالمشروع فيه , و أضغط زر Ok

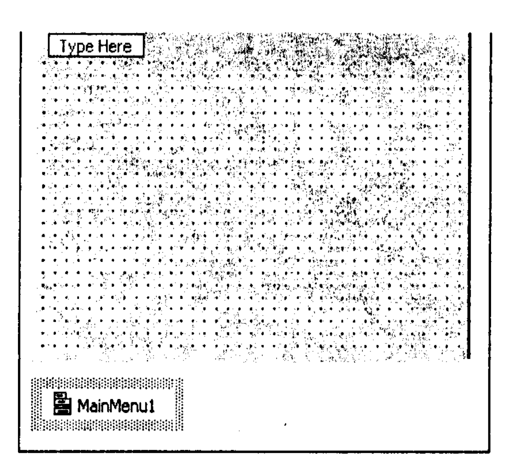


المربع الحواري الخاص بإنشاء مشروع جديد

(شكل رقم ٧-٤)

بالضغط علي زر Ok يتم إغلاق المربع الحواري و يقوم فيجوال أستوديو بإنشاء الملفات الخاصة بالمشروع الجديد في المجلد الذي اخترته , و يتم إنشاء نموذج فارغ في نافذة المشروعات في وسط واجهة البرنامج , قم بالنقر فوق خاصية MainMenu في شريط الأدوات و قم برسم مربع في النموذج الفارغ الذي قام البرنامج بإنشائه كالموضح في (شكل رقم ٧-٥) ليمثل هذا المربع القائمة .





المربع الذي يمثل القائمة

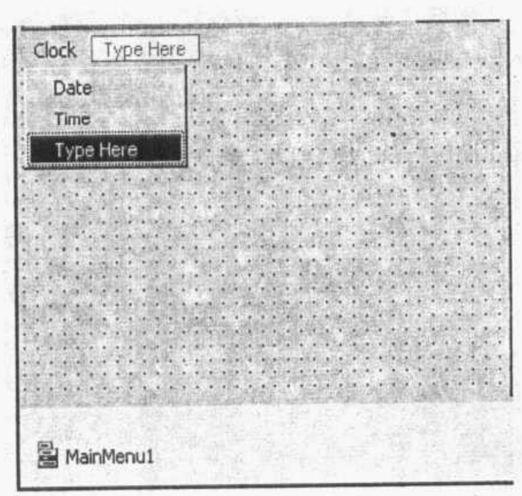
(شكل رقم ٧-٥)

و في الشكل السابق بإنشاء المربع الذي يمثل القائمة تظهر عبارة Type Here و التي تعني أنه يمكنك مباشرة كتابة اسم القائمة في هذا المكان , و بعد كتابة أسم القائمة يمكنك بالنقر علي السهم الأسود الصغير الموجود بجوار الاسم يمكنك كتابة أسماء الأوامر التي تحتوي عليها القائمة أو كتابة أسماء الأوامر الفرعية التي تحتوي عليها هذه القائمة , كما يمكنك تعديل أسماء القوائم و الأوامر التي تحتوي عليها أو القوائم الفرعية بعد ذلك من خلال Line Menu Designer كما يمكنك الإضافة أو الطرح من هذه القوائم .

في المربع الذي يمثل القائمة الذي قمت برسمه كما يوضحه الشكل السابق , أنقر فوق كلمة Type Here و قم بكتابة كلمة Clock و أضغط مفتاح Enter في لوحة المفاتيح لظهر بذلك السهم الأسود بجوار القائمة و بالنقر فوق

هذا السهم يظهر المزيد من عبارات Type Here و التي تتيح لك كتابة الأوامر التي تحتوي عليها القائمة التي قمت بإنشائها أو القوائم الفرعية التي تحتوي عليها هذه القائمة .

في المكان المخصص لكتابة أسماء الأوامر التي تحتوي عليها القائمة أكتب Date و أضغط مفتاح Enter في لوحة المفاتيح لتحديد أول أمر في القائمة , ثم في الحقل الثاني أكتب كلمة Time و أضغط مفتاح Enter في لوحة المفاتيح لتكون بذلك قد قمت بإضافة أمرين إلي القائمة كما يوضح (شكل رقم ٧-٦) .



إضافة الأوامر إلي القائمة
(شكل رقم ٧-٦)

قم بالنقر في أي مكان فارغ في النموذج لإغلاق محرر الأوامر , و بعد

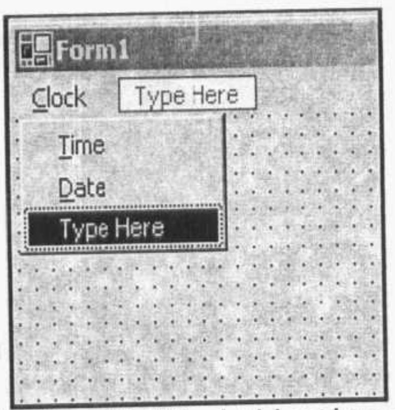


ذلك بالنقر علي كلمة Clock في النموذج ستظهر الأوامر التي تحتوي عليها هذه القائمة .

تخصيص اختصارات لوحة مفاتيح للقائمة

عند تحديد خواص القائمة و تحديد خواص الأوامر التي تحتوي عليها هذه القائمة يتيح لك فيجوال أستوديو أيضاً تخصيص اختصارات لوحة مفاتيح لتنفيذ الأوامر الموجودة في هذه القائمة .

و لتخصيص اختصارات لوحة المفاتيح لتنفيذ الأوامر التي تحتوي عليها القائمة قم بالنقر فوق قائمة Clock في النموذج , و إذا قمت بالنقر مرة أخري فوق أسم القائمة يظهر هذا الاسم علي النحو الذي تستطيع معه تعديل هذا الاسم , أنقر مرة أخري فوق أسم القائمة لظهر أسم القائمة علي النحو الذي تستطيع معه تعديل الاسم , قم بالضغط علي مفتاح السهم الأيمن في لوحة المفاتيح ليتحرك مؤشر الكتابة عند حرف C الموجود في أسم القائمة Clock ثم أكتب علامة & لتحديد الحرف C كاختصار لوحة مفاتيح لفتح هذه القائمة و أضغط مفتاح Enter في لوحة المفاتيح , ثم قم بالانتقال إلي أسماء الأوامر التي تحتوي عليها القائمة و كرر نفس الشيء مع هذه الأوامر ليكون حرف D هو مفتاح الاختصار الخاص بالأمر D و الحرف T هو مفتاح الاختصار للأمر Time كما يوضحها (شكل رقم ٧-٧) .



تحديد اختصارات لوحة المفاتيح للقائمة
(شكل رقم ٧-٧)

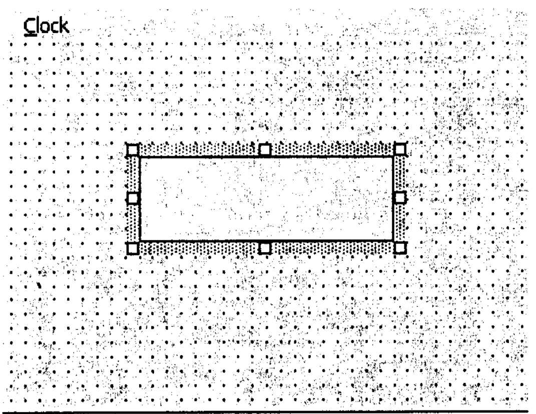
تغيير ترتيب الأوامر في القائمة

لتغيير ترتيب الأوامر في القائمة قم بالنقر علي القائمة في النموذج لتظهر عناصر القائمة , و بظهور عناصر القائمة يكون تغيير ترتيب الأوامر في القائمة في غاية السهولة بمجرد سحب هذه الأوامر في القائمة إلي المكان الذي تريده في القائمة , في القائمة قم بسحب أمر Time في اعلي القائمة , ثم قم بالنقر في أي مكان في النموذج خارج القائمة للخروج من تعديل القائمة .

إضافة مربع عنوان إلي النموذج

لإضافة مربع عنوان إلي النموذج , قم بالنقر فوق Lable في شريط الأدوات ثم قم بالسحب في النموذج لعمل مربع عنوان كالموضح في

(شكل رقم ٧-٨) .



إضافة مربع عنوان إلي النموذج

(شكل رقم ٧-٨)

من قائمة الخواص قم بتحديد الخواص الموضحة في الجدول التالي لمربع لعنوان الموضح في الشكل السابق .

الخاصة	القيمة
BorderStyle	FixedSingle
Font	Microsoft San Serif , Bold , 14 –point
Text	Empty
TixtAlign	MiddleCneter

جدول يوضح خواص مربع العنوان

تحديد خواص الأوامر

لتحديد خواص الأوامر التي تحتوي عليها القائمة لتحديد ما يحدث عندما يقوم المستخدم باختيار هذه الأوامر من القائمة , قم بالنقر علي القائمة

Clock في النموذج لفتحها , و أنقر نقراً مزدوجاً علي أمر Time ليتم فتح محرر الأكواد و يظهر به الأكواد الخاصة بتحديد خواص هذا الأمر .

في محرر الأكواد و بين السطرين الأول و الأخير الخاصين بخواص الأمر قم بكتابة الكود التالي , و أضغط مفتاح السهم الأسفل في لوحة المفاتيح

```
● Label1.Text = TimeString
```

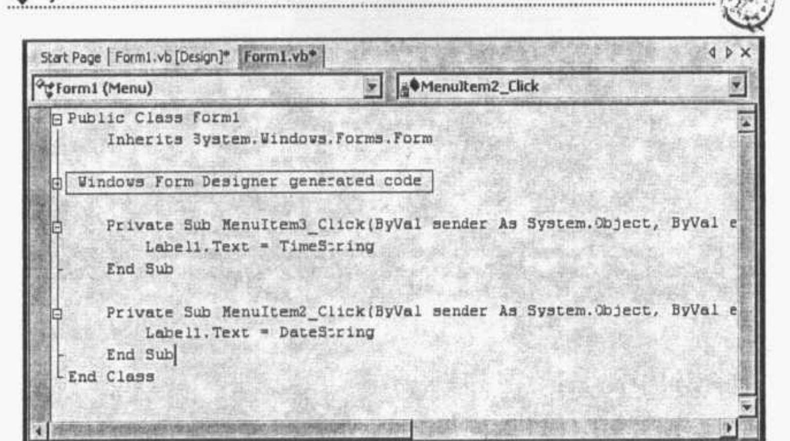
و معني الكود السابق أنه في حالة قيام المستخدم باختيار هذا الأمر من القائمة يتم عرض الوقت الحالي في مربع العنوان Lable1 الموجود في النموذج , و يقوم هذا الكود أيضاً بتعريف المتغير Time و الذي يحتوي علي قيمة الوقت الحالي علي أنه متغير لفظي .

ثم قم بالعودة إلي نمط عرض التصميم في واجهة برنامج فيجوال أستوديو , و أنقر نقراً مزدوجاً فوق أمر Date في القائمة التي قمت بإنشائها , و في محرر الأكواد و بين السطرين الخاصين بهذا الأمر قم بكتابة الكود التالي , و أضغط مفتاح السهم لأسفل في لوحة المفاتيح .



```
● Label1.Text = DateString
```

بحيث بعد كتابة هذا الكود تصبح الأكواد الموجودة في محرر الأكواد و الخاصة بخواص الأمر Date كما يوضح (شكل رقم ٧-٩) .

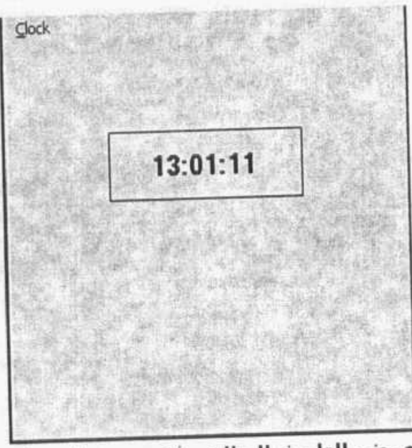


الأكواد الموجودة في محرر الأكواد و الخاصة بالأمر Date

(شكل رقم ٧-٩)

و هذا الكود معناه أن يتم عرض التاريخ الحالي في مربع العنوان Lable1 عندما يقوم المستخدم باختيار أمر Date من القائمة .

أضغط زر الحفظ Save All لحفظ المشروع , ثم قم بالضغط علي زر التشغيل لتجربة المشروع في واجهة البرنامج , و قم باختيار Date من القائمة ليظهر بذلك التاريخ في مربع العنوان كما يوضح (شكل رقم ٧-١٠)



عرض التاريخ الحالي في مربع العنوان

(شكل رقم ٧-١٠)

مربعات الحوار Dialog Boxes

يحتوي برنامج فيجوال أستوديو يحتوي علي سبعة أنواع من مربعات الحوار لذلك فيمكنك استخدام أي من هذه الأنواع علي حسب حاجتك و قد لا تحتاج إلي إنشاء مربع حوار خاص بك و كل ما عليك هو ربط المربع الحواري بأحد الأوامر الموجودة في القائمة , و الجدول التالي يوضح أنواع مربعات الحوار السبعة المتوافرة في البرنامج .

أسم مربع ا حوار	وظيفته
OpenFileDialog	فتح ملف
SaveFileDialog	حفظ التعديلات التي أجريت علي ملف

أسم مربع ا حوار	وظيفته
FontDialog	تحديد خواص الخط من حيث النوع و الحجم و التنسيق
ColorDialog	فتح بالتة الألوان لاختيار الألوان
PrintDialog	يحتوي علي خيارات الطباعة
PrintPreviewDialog	معاينة قبل الطباعة
PageSetupDialog	أعدادات الصفحة

جدول يوضح أنواع مربعات الحوار الموجودة في برنامج فيجوال أستوديو

و في التطبيق التالي سوف نوضح كيفية استخدام مربعات الحوار الموجودة في البرنامج و ربط هذه المربعات بأوامر القائمة .

فتح المشروع

في هذا التطبيق سوف نقوم بفتح التطبيق السابق Clock و الذي قمنا بتنفيذه في الجزء الأول من هذا الفصل ثم سنقوم بإضافة مربعات حوار إلي عناصر القائمة التي أنشأناها في المشروع .

لفتح المشروع أختار Open من قائمة File و منها اختار Project و تصفح جهازك حتى تصل إلي المشروع الخاص بالقوائم الذي قمت بإنشائه في الجزء السابق من هذا الفصل و أضغط زر Open ليتم فتح المشروع في واجهة البرنامج .

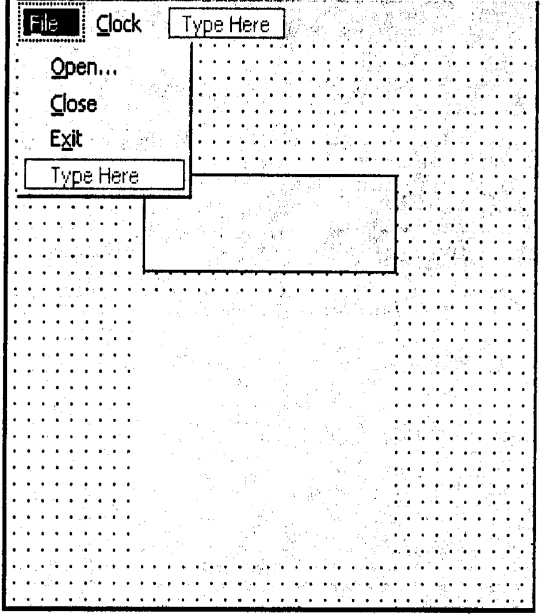


إضافة مكونات إلي المشروع

فيما يلي سوف نقوم بإضافة مربع صورة و قائمة أخري تسمي File إلي المشروع الذي قمت بفتحه .

- لإضافة مربع صورة اختار PictureBox من شريط الأدوات و قم بالسحب في وسط النموذج لإضافة مربع صورة كما يوضح (شكل رقم ٧-١١) , و من قائمة الخواص قم بضبط خاصية SizeMode عند القيمة StretchImge
- و لإضافة قائمة File إلي المشروع , أنقر فوق قائمة Clock الموجودة في المشروع , ثم قم بالنقر فوق كلمة Type Here و اكتب كلمة &File لإنشاء قائمة File و تخصيص مفتاح الاختصار F في لوحة المفاتيح لفتح هذه القائمة , ثم أضغط مفتاح السهم لأسفل في لوحة المفاتيح و أكتب كلمة &Open لعمل أمر Open و يكون مفتاح الاختصار الخاص به في لوحة المفاتيح هو حرف O ثم أضغط مفتاح السهم الأسفل في لوحة المفاتيح و اكتب كلمة &Close لعمل أمر Close و يكون مفتاح الاختصار الخاص بهذا الأمر في لوحة المفاتيح هو حرف C ثم أضغط مفتاح السهم لأسفل في لوحة المفاتيح و اكتب كلمة E&xit لعمل أمر Exit و تخصيص حرف X في لوحة المفاتيح لتشغيل هذا الأمر .
- قم بعد ذلك بسحب قائمة File فوق قائمة Clock حتى تصبح هذه القائمة بجوارها في شريط القوائم كما يوضح الشكل .





إضافة قائمة File و مربع صور إلي المشروع السابق

(شكل رقم ٧-١١)

تغيير أسم القائمة

كما سبق و أن ذكرنا أنه يمكنك تغيير أسماء العناصر التي يحتوي عليها النموذج مثل الأزرار و القوائم و محتويات هذه القوائم و الحقول النصية و أي عنصر أخر يحتوي عليه النموذج , و لكن بتغيير أسم أي عنصر من العناصر التي يحتوي عليها النموذج لا يتم تغيير أسم هذا العنصر في نافذة الخواص بل يظل محتفظاً بالاسم القديم و أنما ما يتغير فقط هو الاسم

النهائي للعنصر الذي يظهر للمستخدم للمشروع في شكله النهائي .

و لتغيير أسم قائمة File ليصبح mnuFile قم بالنقر علي قائمة File في النموذج و قم بفتح نافذة الخواص Properties Window و في هذه النافذة قم بتغيير أسم القائمة ليصبح mnuFile و بنفس الطريقة قم بتغيير أسم عناصر القائمة , قم بتغيير أسم الأمر Open ليصبح mnuOpenItem و تغيير أسم الأمر Close ليصبح mnuCloseItem و تغيير أسم الأمر Exit ليصبح mnuExitItem ثم أضغط زر Save All لحفظ كل التعديلات التي قمت بإجرائها علي عناصر النموذج .

تعطيل أمر Close

في كل البرامج لا تكون كل الأوامر التي تحتوي عليها ألقوائم نشطة في كل الأوقات بل أن الأمر يجب أن لا يكون نشطاً إلا في حالة الاحتياج إليه فعلي سبيل المثال الأمر Close الخاص بإغلاق الملفات لا يجب أن يكون هذا الأمر نشطاً إلا في حالة ما إذا كان هناك ملف مفتوح بالفعل و يريد المستخدم إغلاقه باستخدام هذا الأمر و فيما عدا ذلك يكون الأمر غير نشط أو معطل في القائمة , و بجعل الأمر غير نشط في القائمة يظهر الأمر معتماً , و بتنشيطه في حالة فتح ملف و نريد استخدام هذا الأمر في إغلاق الملف يكون الأمر واضحاً و نشطاً .

و لتعطيل أمر Close الموجود في قائمة File قم بالنقر علي الأمر في القائمة لتنشيطه , و قم بفتح نافذة الخواص Properties Window و في هذه



النافذة قم بتغيير قيمة خاصية Enabled إلي القيمة False

إضافة أمر إلي قائمة Clock

قم بالنقر فوق قائمة Clock ثم قم بالنقر فوق كلمة Type Here و قم بكتابة الأمر التالي Text Co&lor لإضافة أمر Text Color إلي القائمة و تخصيص حرف L في لوحة المفاتيح كاختصار لاستخدام هذا الأمر .

مربعات الحوار

لإظهار مربع حواري معين من بين الأنواع المختلفة لمربعات الحوار الموجودة في البرنامج السابق الإشارة إليها يجب كتابة أسم مربع الحواري الذي تريد إظهاره باختيار أحد الأوامر الموجودة في القائمة ضمن جملة ShowDialog المصاحبة للأمر الذي تقوم بتعديل الكود الخاص به في محرر الأكواد , و فيما يلي سوف نتعرف علي الأكواد الخاصة بتحديد خواص الأوامر التي تحتوي عليها القوائم و كيفية استخدام هذه الأكواد في فتح مربعات حوارية مع قيام المستخدم باختيار أوامر معينة من القائمة .

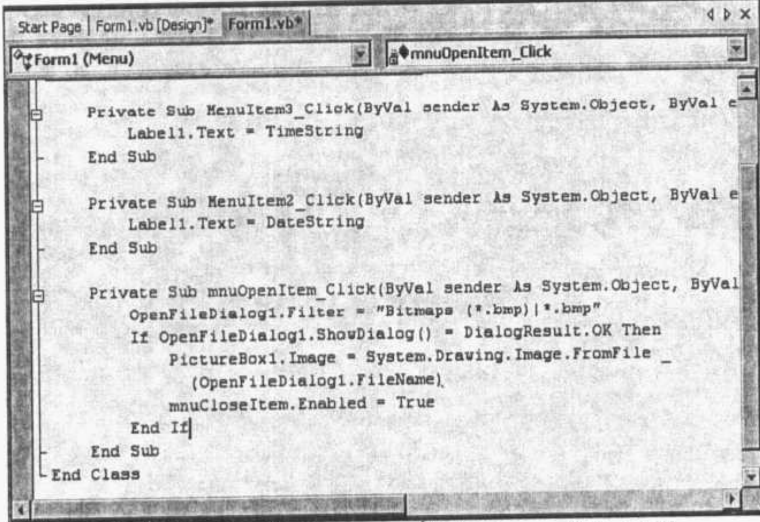
أمر Open

أنقر نقراً مزدوجاً فوق أمر Open الموجود في قائمة File في النموذج لفتح محرر الأكواد و به الأكواد الخاصة بتشغيل هذا الأمر , و بين السطرين الخاصين بوظيفة هذا الأمر قم بكتابة الكود التالي , و أضغط مفتاح السهم الأسفل في لوحة المفاتيح .

- OpenFileDialog1.Filter = "Bitmaps (*.bmp)|*.bmp"

- If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
- PictureBox1.Image = System.Drawing.Image.FromFile _
- (OpenFileDialog1.FileName)
- mnuCloseItem.Enabled = True
- End If

بحيث يكون محرر الأكواد بعد كتابة هذا الأمر علي النحو الموضح في (شكل رقم ٧-١٢) .



الكود الخاص بوظيفة الأمر Open في محرر الأكواد

(شكل رقم ٧-١٢)

و في الكود السابق تشير الأسطر الثلاثة الأولي من الكود إلي خواص المربع الحواري الذي يتم فتحه بقيام المستخدم باختيار هذا الأمر من القائمة , و الجملة الأولي تعتبر فلتر Filter لتحديد أنواع الملفات التي يمكن أن يتم فتحها بواسطة المربع الحواري الخاص بفتح الملفات المصاحب لأمر Open و

يمكنك إضافة أنواع جديدة للملفات التي يمكن فتحها بواسطة هذا المربع الحواري عن طريق إضافة امتداد أخري من الملفات إلي المربع الحواري بالكود التالي :

```
● OpenFileDialog1.Filter = "Bitmaps (*.bmp)|*.bmp|Metafiles (.wmf)|
  .wmf"
```

أمر Close

قم بالخروج من محرر الأكواد و العودة إلي نمط عرض التصميم بالنقر علي الزر الخاص بذلك , و أنقر نقراً مزدوجاً فوق أمر Close في قائمة File ليظهر محرر الأكواد مرة أخري و به أول و أخر سطر من الجملة البرمجية الخاصة بهذا الأمر في القائمة , و بين هذين السطرين قم بكتابة الكود التالي :

```
● PictureBox1.Image = Nothing
● mnuCloseItem.Enabled = False
```

و الكود السابق يقوم بإغلاق الملف المفتوح في مربع الصورة , و يقوم بتفريغ هذا المربع .

أمر Exit

لتحديد وظيفة أمر Exit قم بالانتقال إلي نمط عرض التصميم مرة أخري و أنقر نقراً مزدوجاً فوق أمر Exit في قائمة File ليظهر بذلك محرر الأكواد و يعرض السطر الأول و الأخير من الجملة البرمجية الخاصة بوظيفة هذا الأمر , و بين هذا السطرين قم بكتابة الأمر التالي :

```
● End
```

و بذلك يتم الخروج من البرنامج باختيار هذا الأمر من القائمة .

أمر Text Color

لتعديل أمر Text Color الذي قمت بإضافته إلي قائمة Clock قم بالخروج من محرر الأكواد و العودة إلي نمط التصميم , و في النموذج أنقر فوق هذا الأمر نقراً مزدوجاً في القائمة ليتم بذلك فتح محرر الأكواد و يحتوي علي الجملة الخاصة بوظيفة هذا الأمر , بين السطر الأول و الأخير من الجملة البرمجية الخاصة بالأمر قم بكتابة الكود التالي :

```
• ColorDialog1.ShowDialog()
• Label1.ForeColor = ColorDialog1.Color
```

و يمكن استخدام المربع الحواري Color الذي يتم فتحه باختيار هذا الأمر من القائمة في التحكم في ألوان أي عنصر من واجهة البرنامج الذي تقوم بتصميمه أو لون خلفية البرنامج .

و أول سطر في الكود السابق يقوم بإظهار المربع الحواري الخاص بالألوان باستخدام جملة ShowDialog و تستخدم هذه الجملة لفتح أي مربع حواري تقوم بتحديد أسمه في الكود مرتبط بعنصر التحكم الذي تقوم بكتابة الكود الخاص بتحديد وظيفته , و السطر الثاني من الكود متعلق بتخصيص الأمر الذي يقوم المستخدم باختياره من مربع الألوان لمربع العنوان Lable1

تحديد خواص مربع الألوان

يمكنك أيضاً التحكم في خواص مربع الألوان الذي يظهر باختيار أمر Text Color من القائمة و ذلك من خلال نافذة الخواص Properties

Window أو يمكنك عمل ذلك أيضاً باستخدام الأكواد , و الجدول التالي يوضح وظائف أهم الخيارات الموجودة في نافذة الخيارات و المتعلقة بمربع الألوان .

الخاصية	وظيفتها
AllowFullOpen	بتحديد قيمة هذه الخاصية عند True يظهر في المربع الحواري الزر الخاص بإضافة ألوان إلي الزر
AnyColor	بتحديد قيمة هذه الخاصية عند True تمكن المستخدم من اختيار أي لون موجود في مربع الألوان
FullOpen	بتحديد قيمة هذه الخاصية عند القيمة True يقوم بإظهار المنطقة الخاصة بالألوان الإضافية في مربع الألوان
ShowHelp	بضبط قيمة هذه الخاصية عند True تؤدي إلي ظهور الزر الخاص بالتعليمات في مربع الألوان
SolidColorOnly	بضبط هذه الخاصية عند القيمة True لا يستطيع المستخدم الاختيار سوي من بين الألوان الأساسية فقط في مربع الألوان

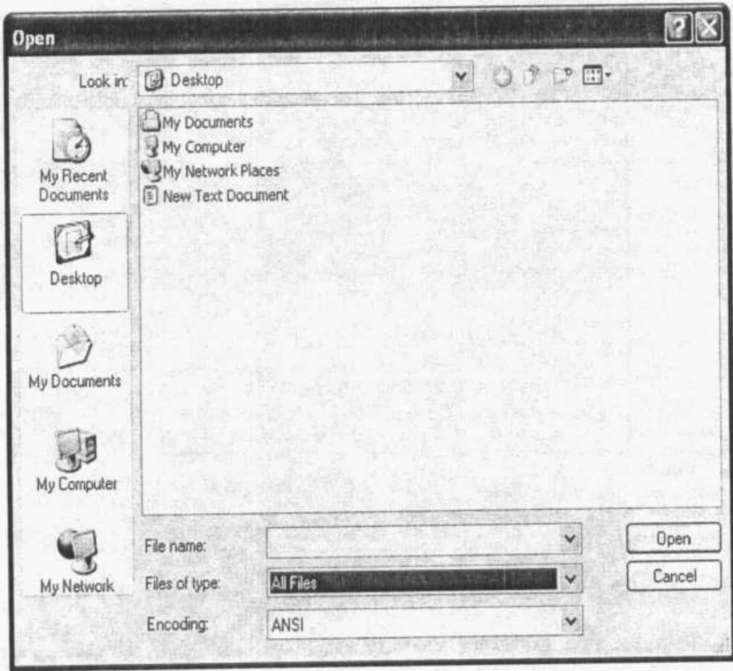
جدول يوضح الخواص المتعلقة بمربع الألوان في نافذة الخواص

تشغيل البرنامج

لتشغيل البرنامج قم بالنقر فوق زر Start الموجود في شريط الأدوات الرئيسي , و في نافذة البرنامج أختار Open من قائمة File ليظهر بذلك



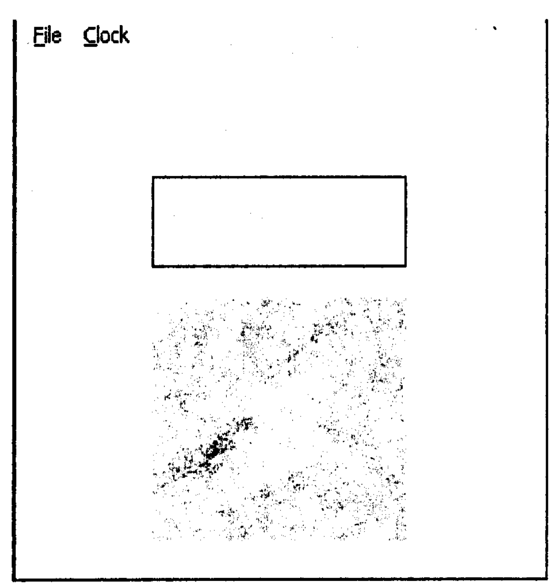
المربع الحواري الخاص بفتح الملفات كما يوضحه (شكل رقم ٧-١٣) و المرتبط ظهورة باختيار هذا الأمر من القائمة .



المربع الحواري الخاص بفتح الملفات
(شكل رقم ٧-١٣)

و من المربع الحواري الخاص بفتح الملفات الموضح في الشكل السابق أختار أي صورة bitmap و أضغط زر Open ليتم فتح هذه الصورة في مربع الصورة الموجود في النموذج كما يوضح (شكل رقم ٧-١٤) .



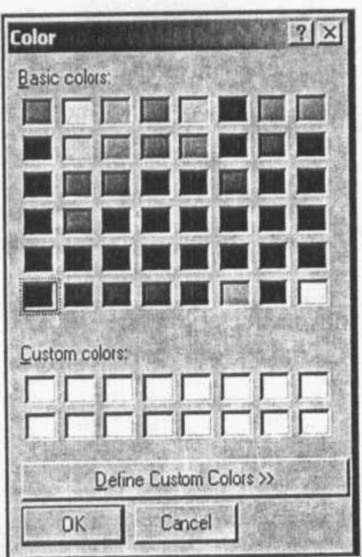


فتح الصورة في مربع الصورة

(شكل رقم ٧-١٤)

و باختيار أمر Time من قائمة Clock يظهر التاريخ الحالي في مربع العنوان الموجود في النموذج , و باختيار أمر Text Color من نفس القائمة يظهر مربع الألوان كما يوضحه (شكل رقم ٧-١٥) .

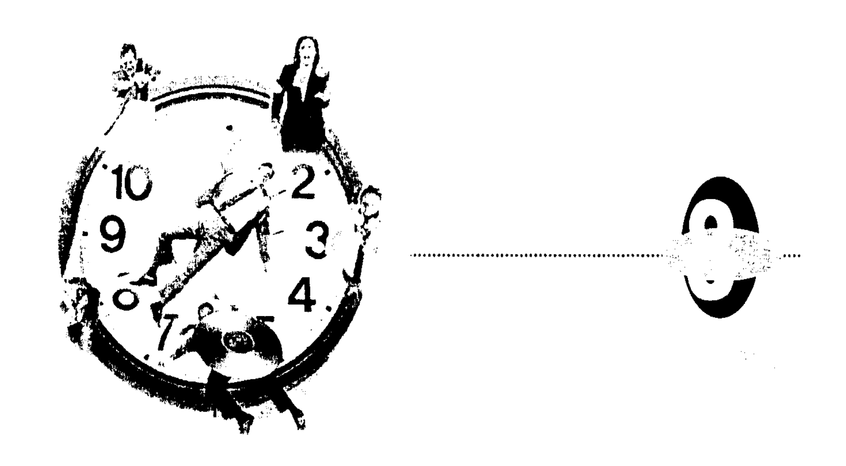




مربع الألوان الذي يظهر باختيار أمر Text Color

(شكل رقم ٧-١٥)

و من مربع الألوان الموضح في الشكل السابق باختيار أي من هذه الألوان و الضغط علي زر Ok يتم تلوين النص الموجود في المربع الموجود في النموذج بهذا اللون و يتم إغلاق مربع الألوان



التعامل مع قواعد البيانات

تعلم بنفسك Visual Basic.net 2005

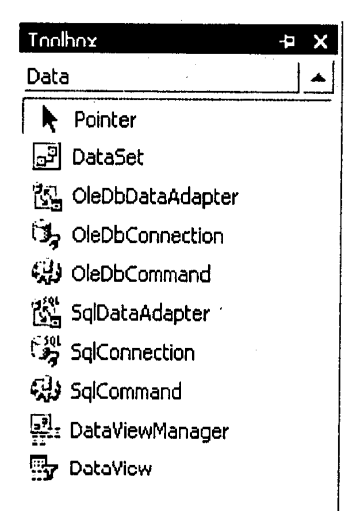
التعامل مع قواعد البيانات

من المزايا الهامة لفيجوال بيسيك دوت نت أمكانية الوصول إلي قواعد البيانات و استخراج البيانات من قواعد البيانات , و التعامل مع البيانات الموجودة في قواعد البيانات .

استخدام النماذج في الوصول إلي قواعد البيانات

يمكنك أنشاء مشروع يحتوي علي نموذج كالمشروعات التي قمنا بإنشائها في الفصل السابق , و إضافة عناصر إلي هذا النموذج تمكن من الوصول إلي قواعد البيانات , فلو قمت بفتح نافذة الأدوات تلاحظ أن هذه النافذة تحتوي علي شريط أدوات بأعلى هذه النافذة يسمي Data و بالنقر فوق هذا الشريط ستشاهد مجموعة من أدوات التحكم التي تستخدم في التعامل مع قواعد البيانات كما يوضحها (شكل رقم ٨-١) .

و يمكنك استخدام أدوات التعامل مع قواعد البيانات هذه التي تظهر في شريط الأدوات الموضح في الشكل عن طريق سحب هذه الأدوات بزر الماوس من شريط الأدوات إلي النموذج الذي تقوم بإنشائه مباشرة .



مجموعة الأدوات التي تستخدم في التعامل مع قواعد البيانات

(شكل رقم ٨-١)

كما يمكنك أيضاً التعامل مع قواعد البيانات في فيجوال بيسيك دوت نت من خلال الأكواد التي تقوم بكتابتها , فيوجد فئات معينة Classes يمكن استخدامها مع أكثر من نموذج في وقت واحد للتعامل مع قواعد البيانات , فيمكن استخدام عنصر واحد في المشروع كله للتعامل مع قواعد البيانات و تقوم كل النماذج التي يتكون منها المشروع باستخدام هذا العنصر في التعامل مع قواعد البيانات بدلاً من وضع عناصر تحكم في كل نموذج خاصة بالتعامل مع قواعد البيانات .

و لعمل عنصر للتعامل مع قواعد البيانات اختار Add New Item من قائمة Project ثم أنقر نقراً مزدوجاً فوق أيقونة Component Class ليتم إضافة

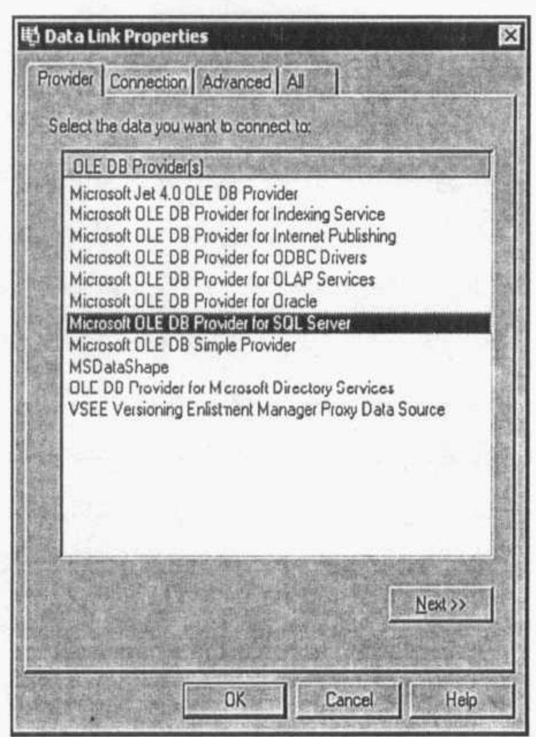



عنصر للتعامل مع قواعد البيانات إلي المشروع يمسي Component1

استخدام معالج التعامل مع قواعد البيانات

لفتح معالج التعامل مع قواعد البيانات قم بفتح صندوق الأدوات ToolBox ثم اختار شريط أدوات التعامل مع قواعد البيانات Data Tab من هذا الصندوق , و قم بسحب أداة OleDbDataAdapter إلي النموذج , ليتم بذلك فتح نافذة معالج التعامل مع قواعد البيانات , و في أول نافذة للمعالج قم بالنقر فوق زر Next لتظهر بذلك النافذة الثانية للمعالج لتطلب منك اختيار أي اتصال بقواعد البيانات تريد استخدامه في التعامل مع قواعد البيانات , و في حالة ما لم يكن لديك اتصال سابق بقواعد البيانات قم بالنقر فوق زر New Connection ليظهر بذلك المربع الحواري الخاص بإنشاء اتصال جديد بقواعد البيانات كما هو موضح في (شكل رقم ٨-٢) .



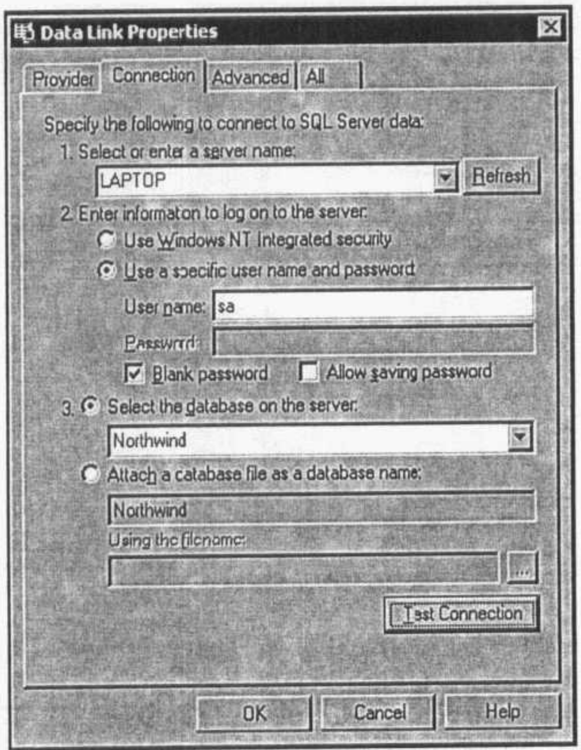


المربع الحواري الخاص بإنشاء اتصال بقواعد البيانات

(شكل رقم ٨-٢)

في المربع الحواري الخاص بإنشاء اتصال بقواعد البيانات الموضح في الشكل السابق أنقر فوق زر Connection و أكتب أسم سيرفر قواعد البيانات الذي تريد الاتصال به في حقل Server Name كما يوضح (شكل رقم ٨-٣)

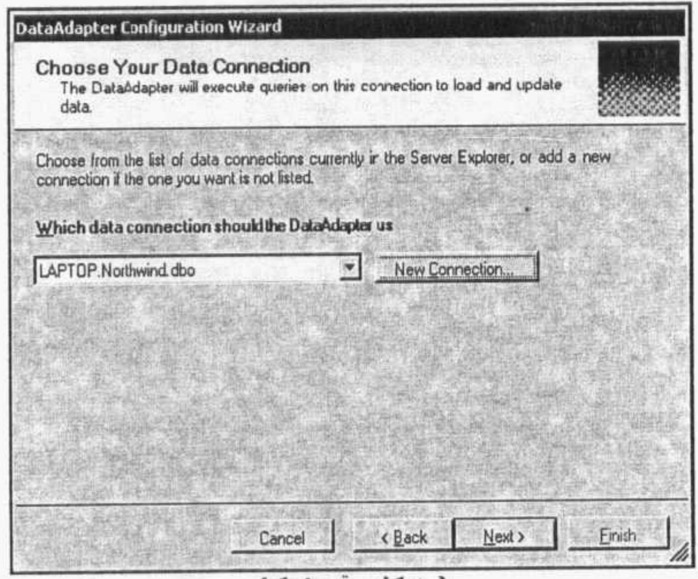




المربع الحواري الخاص باختيار سيرفر قواعد البيانات

(شكل رقم ٨-٣)

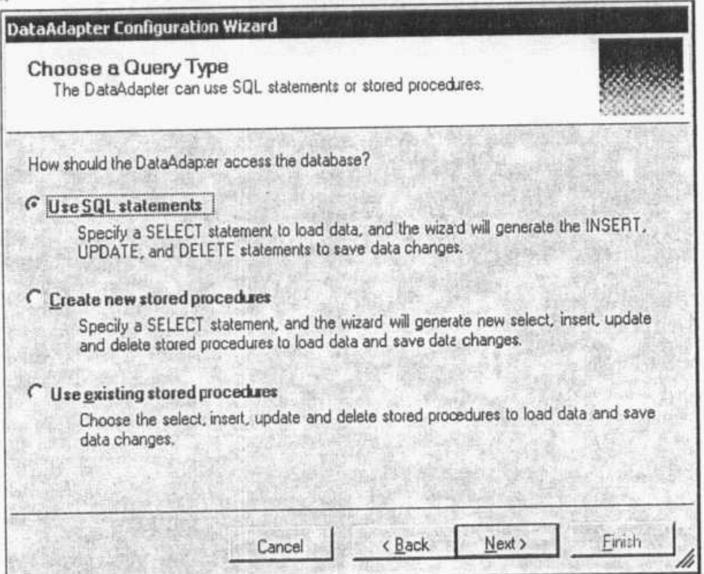
و باختيار سيرفر قواعد البيانات من المربع الحواري الموضح في الشكل السابق و الضغط علي زر Ok تظهر النافذة الموضحة في (شكل رقم ٨-٤) , في معالج الاتصال بقواعد البيانات .



(شكل رقم ٨-٤)

في النافذة الموضحة في الشكل السابق أضغط زر Next لتظهر بذلك النافذة الموضحة في (شكل رقم ٨-٥) و التي تطلب منك تحديد ما سوف تقوم بإجرائه مع قواعد البيانات بعد اختيار الاتصال بقواعد البيانات , أي تحديد ما هي العملية التي تريد أن تقوم بإجرائها مع قواعد البيانات , مثل استخراج بيانات أو وضع بيانات أو التعامل مع البيانات نفسها الموجودة في قواعد البيانات .

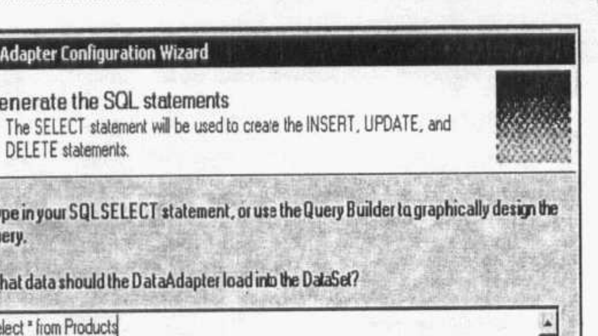


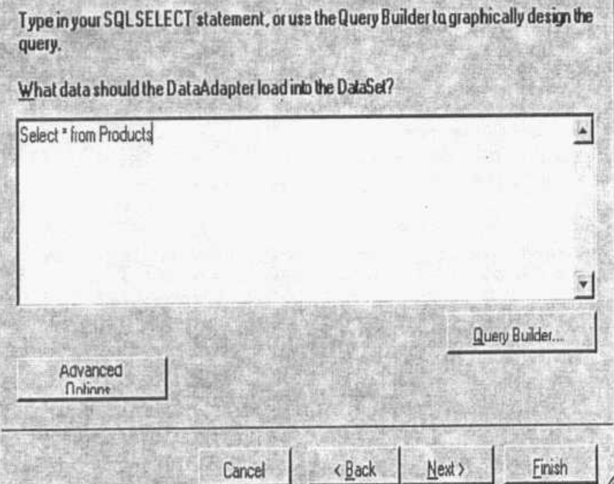


النافذة الخاصة بتحديد نوع التعامل مع قواعد البيانات

(شكل رقم ٨-٥)

و من النافذة الخاصة بتحديد نوع العملية التي تريد أجراؤها مع قواعد البيانات الموضحة في الشكل السابق باختيار Use SQL Statements و هو الخيار الأول في النافذة و الضغط علي زر Next تظهر النافذة الخاصة بكتابة جملة بلغة SQL الخاصة بالتخاطب مع قواعد البيانات , كما يوضحها (شكل رقم ٨-٦) .





النافذة الخاصة بكتابة جملة بلغة SQL

(شكل رقم ٨-٦)

في النافذة الخاصة بكتابة جملة بلغة SQL للتخاطب مع قواعد البيانات الموضحة في الشكل السابق قم بكتابة الجملة التي تريد توجيه الأمر بها لقواعد البيانات باستخدام لغة التخاطب مع قواعد البيانات SQL و أضغط زر Next لتظهر بذلك أخر نافذة في المعالج و التي تحتوي علي مجرد معلومات عن العملية التي قمت بإجرائها مع قواعد البيانات , في هذه النافذة أضغط زر Finish للخروج من المعالج .



و بالخروج من معالج التعامل مع قواعد البيانات بعد أتمام التعامل مع قواعد البيانات تعود لعنصر Component1.vb في نافذة التصميم , و بالنظر في نافذة التصميم تلاحظ عنصرين تحكم جديدين تم إضافتهما إلي النافذة هما العنصر oleDbDataAdapter1 و العنصر oldDconnection1 و اللذين يمثلان التعامل الاتصال و التعامل مع قواعد البيانات قد تم إضافتهما إلي النموذج الموجود في نافذة التصميم , و بالنظر في متصفح المشروع الموجود في يسار واجهة البرنامج تلاحظ أنه قد تم إضافة ملف جديد إلي الملفات التي يتكون منها المشروع و هو الملف dsProducts.xsd

إنشاء مجموعة بيانات

لكتابة كود يقوم بإنشاء مجموعة بيانات قم بفتح محرر الأكواد الذي يحتوي علي الأكواد الخاصة بالنموذج , و في هذا المحرر قم بكتابة الكود التالي .

```
• Public Function FillDataSet() as LearningVBdata.dsProducts
• Dim ds as new LearningVBdata.dsProducts
•.oleDbConnection1.Open()
• Try
• Me.oleDbDataAdapter1.Fill(ds)
• Catch fillException As System.Exception
• msgBox("Caught exception: " & fillException.Message)
• End Try
• Me.oleDbConnection1.Close()
• Return ds
```





- End Sub
- Public Sub UpdateDataSource(ByVal ds AsLearningVBdata.dsProducts)
  oleDbConnection1.Open()
- Dim UpdatedRows As System.Data.DataSet
- Dim InsertedRows As System.Data.DataSet
- Dim DeletedRows As System.Data.DataSet
- UpdatedRows = ds.GetChanges(System.Data.DataRowState.Modified)
- InsertedRows = ds.GetChanges(System.Data.DataRowState.Added)
- DeletedRows = ds.GetChanges(System.Data.DataRowState.Deleted)
- Try
- oleDbDataAdapter1.Update(UpdatedRows)
- oleDbDataAdapter1.Update(InsertedRows)
- A c c e s s i n g a D a t a b a s e f rom a Windows Application 111
- oleDbDataAdapter1.Update(DeletedRows)
- Catch updateException As System.Exception
- msgBox("Caught exception: " & updateException.Message)
- End Try
- Me.oleDbConnection1.Close()
- End Sub

و الكود السابق يستخدم في استخراج البيانات من قواعد البيانات باستخدام الاتصال و التخاطب مع قواعد البيانات في شكل مجموعة بيانات .




المحتويات





الفصل الثانى : واجهة برنامج فيجوال أستوديو

الفصل الثالث : قوائم برنامج Visual Studio

الفصل الرابع : المكونات الأخرى لبرنامج Visual Studio



الفصل الخامس : **أدوات التحكم**



الفصل السادس : قواعد لغة فيجوال بيسيك

الفصل السابع : طرق جمع البيانات من المستخدم

